

BionicLoop IDE Software Technical Appendices

Controlled technical support for the frozen investigational software baseline

Baseline tag	ide-software-freeze-2026-08-21
Baseline commit	91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Generated	2026-09-08 09:06 EDT
Status	Approved technical supporting records for IDE submission

Investigational use only.

Included Documents

1. Appendices Manifest
2. Appendix A00 - Enhanced Documentation Level and FDA Content Crosswalk
3. Appendix A01 - Software Requirements Specification
4. Appendix A02 - Software Design Description
5. Appendix A03 - Risk Analysis
6. Appendix A04 - Software Verification and Validation Plan
7. Appendix A05 - Requirements Traceability Matrix
8. Appendix A06 - Cybersecurity Plan
9. Appendix A07 - Formal Evidence Index
10. Appendix A08 - Software Composition and Dependency Inventory
11. Appendix A08A - Build 843 Dependency Verification
12. Appendix A08B - Build 843 SBOM and Advisory Review
13. Appendix A08C - IDE Cybersecurity Closure Matrix
14. Appendix A10 - Software Version and Configuration Identification
15. Appendix A10A - BionicLoop 1.0 (843) Tester Release Record
16. Appendix A10B - Build 843 Working Hardware Evidence Review
17. Appendix A10C - Build 843 Freeze Delta and Study-Build Designation
18. Appendix A11 - IDE Freeze Execution Report
19. Appendix A12 - Post-Approval Software Change Control
20. Appendix A13 - Sponsor and Nonsoftware IDE Ownership Map
21. Appendix A13A - Clinical Policy Confirmation
22. Appendix A14 - Cloud Telemetry Submission-Scope Decision
23. Appendix A15 - Protocol Fingerstick Consistency Issue
24. Appendix A16 - Clinical Approval of Software IDE Decisions
25. Appendix A17 - Insulin Delivery CSV Data-Management Procedure
26. Appendix A18 - Final Software Approvals

Table of Contents

- Table of Contents
- IDE Software Technical Appendices - Manifest
 - Use of This Package
 - Controlled Appendices
 - Formal Evidence Attachments
 - RTM Packaging Note
 - IFU Packaging Note
- Enhanced Documentation Level and FDA Content Crosswalk
 - Determination
 - Content Crosswalk
 - Approval and Submission Boundary
- Software Requirements Specification (SRS)
 - Revision History
 - Requirement Format
 - System Boundary and Operating Context
 - Runtime and Cadence
 - CGM Input Handling
 - Fingerstick BG Input Handling
 - Pump Handling
 - Meal Announce
 - CGM-Outage Sustained Dosing (BG-Run)
 - State Persistence and Recovery
 - Algorithm Verification Controls
 - Telemetry and Traceability
 - UI Safety and Status
 - Clinical Settings and Pregnancy Configuration
 - User Alerts and Escalation
 - Security and Privacy
 - Traceability Source Mapping
- Software Design Description (SDD)
 - Revision History
 - 1. Design Intent
 - 2. Major Components
 - 3. Runtime Sequence (Nominal)

- 4. Key Design Policies
- 5. Data Persistence
- 6. Requirement Allocation
- 7. Design Constraints
- 8. BG Design and Deferred Considerations
- 9. Simulation Strategy (Medium now, High later)
- Controlled Software Risk Analysis
 - Revision History
 - Method
 - Risk Register
 - Residual Risk Acceptance Notes
 - Residual Probability Rationale
 - Overall Residual Risk Conclusion (Benefit-Risk)
- Software Verification and Validation Plan (SVVP)
 - Revision History
 - 1. Test Document Acronyms
 - 2. Verification Strategy
 - 3. Test Environments
 - 4. Entry and Exit Criteria
 - 5. Seed Test Inventory
 - 6. Evidence
 - 7. Deferred or Unclaimed Validation
 - 8. Xcode Automated UI Testing Strategy
 - 9. UI Automation Verification Mapping
 - 10. Manual Screenshot UI Review Protocol
- Requirements Traceability Matrix (RTM)
 - Revision History
 - Current Exact-Freeze Evidence Boundary
 - SRS Requirement-Control Traceability
 - Canonical IDE-Scope Matrix
 - Current Interpretation Rules
 - Usage
- Cybersecurity Plan (Investigational Build)
 - Revision History
 - 1. Scope
 - 2. Security Objectives
 - 3. System Trust Boundary Summary

- 4. Supplier / Inherited Control Matrix
- 5. BionicLoop-Owned Local Security Controls
- 6. Security Baseline Controls
- 7. Threat Scenarios (Current Focus)
- 8. Cybersecurity Evidence Artifacts
- 9. Current Control and Evidence Matrix
- 10. Current Missing Artifacts / Open Cyber Gaps
- 11. Deferred Authentication Scope Note
- 12. Incident Response Requirements
- 13. Regulatory Alignment
- IDE Formal Evidence Index - 2026-08-21
 - Formal Lane Index
 - RTM Coverage
 - Evidence Classification
- Cybersecurity Dependency Inventory
 - 1. Purpose
 - 2. Scope of This Snapshot
 - 3. Current Dependency Snapshot
 - 4. Current Observations
 - 5. Current Limitations
 - 6. Current Recommendation
- Build 843 CryptoSwift 1.10 Controlled-Source Verification
 - Controlled Source Identity
 - TDD and Compatibility Finding
 - Verification Results
 - Evidence Classification and Disposition
- Build 843 SBOM and Advisory Review
 - Composition Result
 - Dated Advisory Review
 - Operational Follow-Up
- IDE Cybersecurity Closure Matrix
- IDE Software Version and Configuration Identification
 - 1. Frozen Source Baseline
 - 2. Exact-Freeze Verification Inputs
 - 3. Signed Build 843 Archive
 - 4. IDE Submission Build
 - 5. Documentation and Package Identity

- BionicLoop 1.0 (843) Tester Release Record
 - Source Identity
 - Controlled Inputs
 - Archive Identity
 - App Store Connect Upload Record
 - Retained Release Artifacts
 - Distribution and Installation Status
 - Evidence Classification
- Build 843 Working Hardware Evidence Review
 - Purpose And Baseline Boundary
 - Evidence Hierarchy
 - Retained Device-Local Export
 - Deterministic Checker And Operator Correlation
 - Limitations And Promotion Requirements
 - Conclusion
- Build 843 Freeze Delta and Study-Build Designation
 - 1. Executable Delta Assessment
 - 2. Verification Reuse Assessment
 - 3. Study-Build Designation
- IDE Freeze Execution Report - 2026-08-21
 - 1. Scope
 - 2. Controlled Inputs
 - 3. Formal Lane Results
 - 4. Formal Deviations and Companion Status
 - 5. Remaining Physical / Human Work
 - 6. Freeze Control
- Post-Approval Software Change-Control and Regulatory-Classification Procedure
 - 1. Purpose and Authority
 - 2. Required Assessment for Every Proposed Change
 - 3. Classification Categories
 - 4. Study-Relevant Assessment Examples
 - 5. Verification and Approval
 - 6. Version-to-Subject and Version-to-Data Traceability
 - 7. IRB, Labeling, Protocol, and Reporting Assessment
 - 8. Decision Record Template
 - 9. Regulatory References
- IDE Sponsor and Nonsoftware Ownership Map

- Purpose
- Ownership Map
- Submission Control
- Primary Regulatory Sources
- IDE Clinical Policy Confirmation - 2026-08-21
 - Confirmed Decisions
 - Source Evidence
 - Disposition Boundary
- IDE Cloud Telemetry Submission-Scope Decision
 - Issue
 - Approved Scope
 - Decision Record
- IDE Protocol Fingerstick Consistency Issue
 - Historical Protocol Discrepancy
 - Implemented Behavior
 - Verified Final Protocol Direction
 - Required Disposition
- Clinical Approval of Software IDE Decisions - 2026-08-27 and 2026-08-31
 - Approval Statement
 - Approved Decisions
 - Approved Clinical-Documents Change Directions
 - Clarified Official Study Sources
 - Remaining Administrative Action
- Insulin Delivery CSV Data-Management and Chain-of-Custody Procedure
 - 1. Purpose and Scope
 - 2. Roles
 - 3. Collection Preconditions
 - 4. Export and Original Record
 - 5. Integrity and Completeness Review
 - 6. Recovery and Corroboration
 - 7. Transfer, Access, and Retention
 - 8. Approval Record
- Final Software Approvals - September 2026
 - Approved Items
 - IFU Revision 1.17 Direction - 2026-09-04
 - Final Decision-Register Approvals - 2026-09-04 and 2026-09-07
 - Final Device Labeling - 2026-09-04

- Remaining Operational Work

IDE Software Technical Appendices - Manifest

Field	Value
Package	IDE Software Technical Appendices
Status	Approved controlled technical source set; predeployment condition remains identified
Baseline	ide-software-freeze-2026-08-21 / 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Date	2026-09-08

Use of This Package

These appendices provide detailed design-control and verification support for the IDE Software Submission Core. Historical evidence narratives in a controlled source document remain audit history; current execution status is governed by the canonical RTM current-state matrix and Freeze Execution Report.

Controlled Appendices

ID	Artifact	Current identifier/status
A0 0	Enhanced Documentation Level and FDA Content Crosswalk	Enhanced Documentation Level rationale and controlled FDA content map; approved under D11
A0 1	Software Requirements Specification	SRS version 1.82; approved for IDE submission
A0 2	Software Design Description	SDD version 1.84; frozen G7 implementation limits recorded; approved for IDE submission
A0 3	Risk Analysis	Risk Analysis version 1.53; residual risks accepted under D07; RA-023 predeployment condition retained
A0 4	Software Verification and Validation Plan	SVVP version 1.91; evidence accepted under D08; RA-023 coverage limits retained
A0 5	Requirements Traceability Matrix	RTM version 1.86; canonical current 24-row mapping accepted under D08
A0 6	Cybersecurity Plan	Cybersecurity Plan version 1.03; Build 843 dependency controls, SBOM, supportive Scout scope, and accepted cybersecurity boundary recorded
A0 7	Formal Evidence Index - 2026-08-21	Freeze lane index; execution complete and accepted under D08

ID	Artifact	Current identifier/status
A08	Software Composition and Dependency Inventory	Manual inventory and Build 843 exact dependency evidence accepted within D09
A08A	Build 843 Dependency Verification	Passing engineering evidence for the installed, real-Pod-exercised Build 843 archive approved for IDE submission
A08B	Build 843 SBOM and Advisory Review	CycloneDX 1.5 SBOM reconciliation and dated manual license/advisory review accepted within D09
A08C	IDE Cybersecurity Closure Matrix	Study-used cybersecurity controls separated into complete, open, and deferred items
A09	IFU-BL-001, BionicLoop Instructions for Use	Revision 1.17; approved for IDE submission 2026-09-04; supplied separately rather than embedded in the appendix compilation
A10	Software Version and Configuration Identification	Frozen source, algorithm, dependency, toolchain, and document identity
A10A	BionicLoop 1.0 (843) Tester Release Record	Signed archive, successful distribution, installed identity, completed working real-Pod observation, CC-DEV boundary, retained-artifact locator, and 2026-09-03 IDE-submission approval
A10B	Build 843 Working Hardware Evidence Review - 2026-08-25	Recovery ZIP, operator record, and corroborating telemetry reviewed as supporting evidence accepted under D08
A10C	Build 843 Freeze Delta and Study-Build Designation	Executable delta assessment supporting the 2026-09-03 submission-build approval
A11	IDE Freeze Execution Report - 2026-08-21	Engineering execution complete; three non-product deviations closed by controlled companion executions
A12	Post-Approval Software Change-Control and Regulatory-Classification Procedure	Controlled procedure for sponsor/Regulatory approval
A13	Sponsor and Nonsoftware IDE Ownership Map	Ownership/gap map; nonsoftware completion not inferred
A13A	Clinical Policy Confirmation - 2026-08-21	Written confirmation of the applicable fallback and participant-weight policies; clinical policy complete
A14	Cloud Telemetry Submission-Scope Decision	Supportive Scout role and official study-data/safety-report sources clinically approved 2026-08-27

ID	Artifact	Current identifier/status
A1 5	Protocol Fingerstick Consistency Issue	Closed in the final protocol, consent, and IFU document set
A1 6	Clinical Approval of Software IDE Decisions - 2026-08-27 and 2026-08-31	Controlled correspondence record covering seven approved software/IDE statements, official study sources, and approval to implement all eight software-consistency recommendations
A1 7	Insulin Delivery CSV Data-Management and Chain-of-Custody Procedure	ZIP/CSV collection, transfer, and storage approach approved 2026-09-03; detailed operating controls retained here
A1 8	Final Software Approvals - September 2026	Controlled correspondence approving Build 843, IFU revision 1.17, ZIP/CSV operations, RA-023 submission disposition, residual risks, verification evidence, cybersecurity scope, service environment, and final software-package approver

Formal Evidence Attachments

The formal lane directories contain checksums, commands, environment records, test outputs, and result summaries. Raw `.xcresult` bundles and console logs are controlled evidence attachments and are not reproduced in the narrative PDF. Working evidence is excluded from formal claims unless accepted through the sponsor-approved evidence process.

RTM Packaging Note

The external appendix compilation renders the controlled RTM's current 24-row trace linkage together with the authoritative 2026-08-21 Freeze Execution Overlay. It does not reproduce the source RTM's superseded-candidate evidence and status columns. Those fields remain unchanged in the controlled source and source-control history for internal audit use.

IFU Packaging Note

The appendix compilation does not regenerate or embed IFU screenshots. The separately controlled IFU PDF is the labeling attachment. This prevents an appendix build from silently incorporating unapproved screenshot changes.

Enhanced Documentation Level and FDA Content Crosswalk

Field	Value
Device software	BionicLoop investigational automated insulin-delivery controller
Documentation level	Enhanced
Basis	Software failure or latent defect could contribute to incorrect or delayed insulin delivery and serious injury
Baseline	ide-software-freeze-2026-08-21 / 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Reference	FDA, <i>Content of Premarket Submissions for Device Software Functions</i> , June 2023
Status	Enhanced Documentation Level determination approved within the final software package
Date	2026-09-08

Determination

BionicLoop uses glucose and pump feedback to calculate and command insulin delivery. A software defect could contribute to excess insulin, insufficient insulin, or an interruption of automated delivery. Because those outcomes could cause serious injury, the software package uses the FDA Enhanced Documentation Level. This determination does not depend on the size or phase of the proposed investigation.

Content Crosswalk

Enhanced documentation content	Controlled artifact	Package location/status
Documentation-level rationale	This document; IDE Software Submission Core	Core and Appendix A00
Software description and system context	IDE Software Submission Core; IDE Software Device Description for the Frozen Baseline	Core narrative and controlled supporting description
Software requirements	Software Requirements Specification, version 1.82	Appendix A01

Enhanced documentation content	Controlled artifact	Package location/status
Architecture and detailed design	Software Design Description, version 1.84	Appendix A02
Risk management	Controlled Software Risk Analysis, version 1.53; Known Anomalies and Residuals	Appendix A03 and Core
Software verification and validation	Software Verification and Validation Plan, version 1.91; Formal Evidence Index; Freeze Execution Report	Appendices A04, A07, and A11
Requirements/design/risk/test traceability	Requirements Traceability Matrix, version 1.86	Appendix A05
Cybersecurity	Cybersecurity Plan; Cybersecurity Closure Matrix; SBOM and Advisory Review	Appendices A06, A08B, and A08C
Software version and configuration identification	IDE Software Version and Configuration Identification; Build 843 freeze delta and designation form	Appendices A10 and A10C
Software history and unresolved anomalies	Revision histories; Known Anomalies and Residuals	Controlled source documents and Core
Labeling and operating instructions	IFU-BL-001	Appendix A09, supplied as a separate controlled PDF
Post-approval change control	Post-Approval Software Change-Control and Regulatory-Classification Procedure	Appendix A12

Approval and Submission Boundary

The documentation-level determination and final software package are approved. The IDE Submission Decision Register records the closure basis and the accepted RA-023 predeployment condition. Final FDA assembly, IRB records, and other nonsoftware submission elements remain under study-team control.

Software Requirements Specification (SRS)

Status: Frozen-baseline specification approved for IDE submission Version: 1.82 Owner: BionicLoop engineering

Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, final software-package approval, controlled email dated 2026-09-07 EDT Baseline freeze SHA: 91c0e98a9bc9429a0486bebdebf6c7d8dbbe300e Scope: Closed-loop investigational runtime, device integration, and safety-critical UI behavior Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-25	Engineering	Initial controlled draft baseline
0.9	2026-04-06	BionicLoop engineering	Added document-control metadata and disposition fields
0.91	2026-04-08	BionicLoop engineering	Added masked offline-fallback maintenance, disarm, and fallback-review-event requirements
0.92	2026-04-08	BionicLoop engineering	Clarified 20-minute masked-fallback renewal window, deferred maintenance semantics, and recoverable restore context
0.93	2026-04-14	BionicLoop engineering	Added reconciliation-required blocked-state requirements after offline mask expiry
0.94	2026-04-14	BionicLoop engineering	Added reconnect recovery requirements for connected restore/disarm retry and explicit fresh re-arm after successful recovery
0.95	2026-04-14	BionicLoop engineering	Updated reconnect recovery behavior to auto-arm a fresh session after successful masked-fallback recovery while preserving the fallback review trail
0.96	2026-04-14	BionicLoop engineering	Updated masked-fallback feasibility recovery to preserve the existing session/cadence anchor and resume on the current due step without replay when reconnect restore succeeds
0.97	2026-04-14	BionicLoop engineering	Clarified state-persistence wording for same-session unreconciled resume after successful masked-fallback reconnect recovery
0.98	2026-04-15	BionicLoop engineering	Added pod total-delivery baseline, modeled-vs-pump-reported fallback review, and pre-step missing-fallback arm clarifications
0.99	2026-04-15	BionicLoop engineering	Added confirmed/corrected basal-only reconnect replay requirements and clarified ambiguous unreconciled-resume fallback behavior

Version	Date	Author	Summary of Changes
1.00	2026-04-15	BionicLoop engineering	Added dedicated cloud fallback telemetry contract requirements for BionicScout rendering of fallback lifecycle and delivery/replay summary detail
1.01	2026-04-15	BionicLoop engineering	Clarified that confirmed/corrected reconnect replay must persist replayed internal steps into local export telemetry as distinct replay-marked rows
1.02	2026-04-27	BionicLoop engineering	Clarified that fallback replay must use pump-reported delivered-insulin delta and must not substitute modeled expected delivery when pump reconciliation is unavailable
1.03	2026-05-06	BionicLoop engineering	Added safety-track q5 nominal-basal profile persistence, four six-hour fallback-schedule generation, schedule-aware modeled exposure, and fallback profile/schedule cloud telemetry requirements
1.04	2026-06-02	BionicLoop engineering	Reframed confirmed/corrected fallback recovery as pump-delta delivery reconciliation into the live resumed step, tightened pump-evidence credibility, aligned six-hour schedule-refresh cadence, and added repeated no-active-pod critical alerting
1.05	2026-06-04	BionicLoop engineering	Aligned masked-fallback requirements with the current candidate baseline: confirmed/corrected recovery replays bounded missed primary/secondary algorithm steps with CGM=-1, per-step pump-delivery input, and no catch-up pump commands before the resumed live step
1.06	2026-06-08	BionicLoop engineering	Added schedule-weighted allocation of credible same-pod pump-delta replay across missed fallback steps and clarified zero-weight suppression / zero-delta replay behavior
1.07	2026-06-12	BionicLoop engineering	Added generalized issued-bolus attribution and no-guess replay/disposition requirements for meal, correction, and basal bolus commands interrupted by pump communication loss
1.08	2026-06-13	BionicLoop engineering	Hardened issued-dose reconciliation requirements for non-credible delivered-unit evidence, replacement-pod live-step insulin input, and automatic resume blocking during unresolved holds
1.09	2026-06-15	BionicLoop engineering	Updated different/new-pod issued-dose requirement to assumed-delivered replay/live attribution, replacement-pod command allowance, legacy hold nonblocking behavior, and deferred adaptation-forget consideration
1.10	2026-06-17	BionicLoop engineering	Added one-step algorithm/runtime cadence reconciliation before meal-step selection so loop <code>executedStep</code> , pump request step, and algorithm <code>stepTime</code> remain aligned when the algorithm bridge has already consumed the due step.

Version	Date	Author	Summary of Changes
1.11	2026-06-17	BionicLoop engineering	Added explicit user-confirmed pod-replacement escape for unconfirmed meal delivery progress so the app can record assumed-delivered evidence and open pod setup without leaving the UI trapped.
1.111	2026-06-18	BionicLoop engineering	Added retired/expired/no-active-pod fallback recovery disposition: modeled fallback exposure may be recorded as assumed delivered and replayed when the original pod can no longer provide connected reconciliation evidence. (Renumbered 2026-07-10: this row previously duplicated version 1.11.)
1.12	2026-07-08	BionicLoop engineering	Added request-step-keyed single-consumption protection for reconciled issued doses plus discrepancy evidence when later pump observations disagree with already-consumed delivery.
1.13	2026-07-08	BionicLoop engineering	Added persisted issued-dose evidence restoration into the pump adapter so canceled partial boluses cannot be reconstructed as full deliveries after relaunch.
1.14	2026-07-08	BionicLoop engineering	Added one-refresh patience before reachable-pod assumed meal resolution, algorithm-visible meal-exclusion request-step persistence, and cloud/dashboard-only expired-exclusion disposal evidence.
1.15	2026-07-08	BionicLoop engineering	Documented the accepted no-boundary-cancel issued-dose liveness bound: fresh connected DASH status must converge from active bolus delivery to idle before the next 5-minute step for supported automatic bolus sizes; disconnect/expiry paths report unavailable rather than active delivery.
1.16	2026-07-08	BionicLoop engineering	Added Algo2015 forced-open-loop command block behavior with safety-critical alerting and explicit recovery guidance.
1.17	2026-07-08	BionicLoop engineering	Added masked-fallback recovery-attempt liveness: blocked ordinary wakes re-schedule reconnect recovery and triggers arriving during an in-flight attempt are queued instead of dropped after a field-observed trigger-starvation incident.
1.18	2026-07-08	BionicLoop engineering	Bounded pump-evidence windows to measurement time, degraded stale-baseline amount-fitting deltas to corrected pump-anchored replay, narrowed unreconciled resume to genuinely unusable evidence, and required a safety-critical operator alert when an unreconciled resume may drop nonzero fallback insulin after a field-observed 0.10 U disposal incident.
1.19	2026-07-08	BionicLoop engineering	Added single-flight FIFO loop work passes with start-time state loads and monotonic same-anchor step-base persistence so a stale pass cannot regress the persisted base beneath the algorithm counter and trip a false drift block after a relaunch race.

Version	Date	Author	Summary of Changes
1.20	2026-07-08	BionicLoop engineering	Sponsor decision: replaced the step-mismatch dosing block and manual-reset demand with automatic realignment - the algorithm counter is adopted as the step base without re-anchoring, dosing resumes when wall clock reaches it under fallback basal coverage, and an informational auto-clearing note states the resume window. Added SRS-RUN-007 session-continuity requirement (no automatic session kills, ever).
1.21	2026-07-08	BionicLoop engineering	Fallback replay plans carry their authorizing reconciliation status so the post-recovery re-arm event cannot mask them into disposal; alert telemetry carries disposal source codes; and forced open-loop is presented as a self-clearing CGM-dropout pause with informational-first alerting and persistent-only advisory escalation.
1.22	2026-07-08	BionicLoop engineering	Fallback replay pump tuples echo the algorithm's own prior-row ask as unitsRequested (first row keeps the schedule-share proxy): the Algo2015 reconcile gate matches requested-vs-remembered within one mechanical step (0.01 U), so the previous schedule-share proxy silently skipped delivered-insulin corrections whenever the rounded share diverged from the algorithm's rounded ask - characterized against the real C++ across 0.3-2.0 U/hr.
1.23	2026-07-08	BionicLoop engineering	Sponsor design: an issued dose interrupted before fallback activation replays as a pure confirm row on the dead-gap step plus zero-delivery echo sentinels, then fallback rows - replacing the merged-first-fallback-row convention whose ask+share requested side silently failed the 0.01 U reconcile gate (21:08 EDT field incident: 0.05 U confirmation and first 0.02 U share dropped by the C++). No-room overlap keeps the merge with the requested side echoing the ask exactly; the share credits as over-delivery at the request-step position (accepted policy, bounded to one slot share).
1.151	2026-07-08	BionicLoop engineering	Added Algo2015 checkBG fingerstick prompt behavior with dedupe, haptic feedback, BG-button emphasis, and stale-relaunch suppression. (Renumbered 2026-07-10: this row previously duplicated version 1.15.)
1.24	2026-07-09	BionicLoop engineering	Sponsor-approved Home alert presentation rescope (SRS-ALERT-011): severity-prioritized stack with expansion replaces the vertical carousel, and a root-cause suppression matrix hides generic alerts a specific active alert already explains - Home surface only, never safety-critical, Alert Center always complete.
1.25	2026-07-09	BionicLoop engineering	Sponsor UX decision (SRS-MEAL-006): revalidating an open meal composer re-arms in place on a stale observed step when fresh availability allows, and renders blocked messaging inline with selections preserved - the composer is no longer dismissed

Version	Date	Author	Summary of Changes
			and replaced by a second sheet mid-composition. Submit-time blocking unchanged.
1.26	2026-07-09	BionicLoop engineering	Sponsor field finding (SRS-ALERT-011): when root-cause suppression hides active alerts, the Home stack shall show the hidden related-alert count with a direct Alert Center affordance - the bell badge and Home surface may never silently disagree.
1.27	2026-07-10	BionicLoop engineering	Consolidated SRS-ALERT-019 forced-open behavior; aligned SRS-BG-006 with the implemented step-scoped staleness bound and SRS-SEC-001 with the investigational export posture under RA-009; corrected duplicate revision numbers; retired obsolete feasibility wording; and added SRS-ALG-008 algorithm-instance isolation.
1.28	2026-07-14	BionicLoop engineering	Added the clinically approved SRS-OUTAGE-001..006 CGM-outage family, SRS-PUMP-011 pump-command watchdog, and an SRS-ALERT-019 supersession note while the backup-basal bridge is released.
1.29	2026-07-14	BionicLoop engineering	Clinical IFU-review feedback (Camille Powe): SRS-CLIN-013 first-launch defaults (Standard profile, 120 mg/dL target, 75% meal upfront, 40-minute TMAX for Fiasp; defaults never rewrite persisted values) and SRS-CLIN-014 New Participant Reset for reused phones (full participant-scoped wipe + sign-out, refused while armed or pod active), distinct from the same-participant algorithm reset.
1.30	2026-07-14	BionicLoop engineering	Sponsor device-pass feedback: SRS-CLIN-013 amended (empty-store loads return first-launch defaults without persisting - fixes post-reset resurrection of legacy-era defaults; per-mode wizard target defaults Standard 120 / Pregnancy 110; mode shown in the review summary) and SRS-CLIN-014 amended (armed sessions get a guided stop-then-reset flow instead of a refusal).
1.31	2026-07-14	BionicLoop engineering	Aligned SRS-PUMP-010 with the implemented assumed-delivered clinical-policy disposition for unresolved, mismatched, non-credible, or missing-identity issued-dose evidence once a fresh settled post-request pump status exists; scoped SRS-CLIN-012 to Clinical Settings profile changes outside first-launch setup; aligned reset terminology with the shipped Same-Participant Reset label.
1.32	2026-07-15	BionicLoop engineering	SRS-LOG-010 added: build-designated cloud-log profile (Info.plist-baked level/end-date/label; launch-established integration logging session, one run ID per device per profile window, inert past the end date, no participant-facing controls) supporting field integration-evidence collection.

Version	Date	Author	Summary of Changes
1.33	2026-07-16	BionicLoop engineering	Sponsor-directed logging-config consolidation: SRS-LOG-011 added (server-issued remote logging config fetched at launch/foreground; session > remote > default precedence; BionicScout admin-issued only) and SRS-LOG-004 superseded to no-in-app-control (the debug-build threshold and integration-session Settings controls are removed; study logging is profile- or server-configured).
1.34	2026-07-16	BionicLoop engineering	SRS-LOG-011 amended (sponsor UX direction): periodic re-fetch (nominal 30-minute interval) added to the launch/foreground triggers so long-running phones pick up dashboard changes without a relaunch, and Settings shall show the effective logging state as read-only status text (level, expiry, source: study session / study team / default) with no controls.
1.35	2026-07-16	BionicLoop engineering	Sponsor direction ("just use the cloud session"): SRS-LOG-010 superseded pre-ship - the build-baked profile and the entire on-phone integration-session layer are removed; SRS-LOG-011 amended to the sole-lever posture (threshold = active server config else default; legacy session storage ignored) and gains the threshold-exempt <code>remote_logging_config_applied/_cleared</code> cloud markers on change; SRS-LOG-004 reference updated.
1.36	2026-07-16	BionicLoop engineering	SRS-UI-009 added (sponsor request): server-configurable investigational badge - default NOT FOR HUMAN USE when unset (fail-safe), admin-set text (presets or arbitrary, <= 60 chars), explicit blank shows no badge; fetched on the shared SRS-LOG-011 cadence; mirrored on the BionicScout dashboard chrome.
1.37	2026-07-16	BionicLoop engineering	SRS-OUTAGE-002 amended (field catch, sponsor-reported): "CGM outage is active" precisely defined for the countdown surface - blind executed step AND (algorithm offline mode <code>openLoopMode == 2</code> OR fingerstick-fed step OR BG-run in progress via the persisted schedule's last qualifying source). An isolated blind step with the dropout clock intact (relaunch/wake firing ahead of a healthy sensor's reading) is explicitly not an outage - the prior implementation showed a false fingerstick countdown for ~10 minutes after relaunch.
1.38	2026-07-16	BionicLoop engineering	SRS-OUTAGE-006 amended after a live field discovery: release windows shall be quantified on the engaged fallback event at re-arm; the connected-case algorithm-accounting gap was recorded as a known deviation pending the differentially characterized resolution.
1.39	2026-07-17	BionicLoop engineering	SRS-CLIN-015 added after coverage review: plausible weight-entry range 50-500 lbs gates usable clinical configuration; persisted weight is bounded at 500 lbs without upward

Version	Date	Author	Summary of Changes
			fabrication; and the algorithm layer applies a 20-230 kg final clamp whose low bound under-doses.
1.40	2026-07-17	BionicLoop engineering	RA-009 closure package: SRS-SEC-001 amended (file sharing removed; CSV sealed to gated share; export posture decided), SRS-SEC-002 spool protection clause, SRS-LOG-012 added (sequence-numbered envelopes: durable outbox sequence + install/session identity on every upload; immediate markers reserve durable sequences; Scout completeness basis).
1.41	2026-07-17	BionicLoop engineering	SRS-CLIN-014 amended for reset delivery assurance: pre-reset final flush attempt with an undelivered-count consent line, best-effort <code>telemetry.outbox.discarded</code> marker, sequence-continuity-preserving erasure, no discard on blocked resets, and a deduplicated operator alert for permanent upload-rejection episodes.
1.42	2026-07-18	BionicLoop engineering	SRS-OUTAGE-006 amended after a field-observed pod-away gap was credited as one reconnect tuple: released-window accounting now uses live per-step crediting, and pod-away gaps use standard bounded schedule-weighted replay with baseline re-anchor, exactly-once accumulation, zero-delta retraction rows, identity-mismatch suppression, and pump-measured confirmation when credits accumulate.
1.43	2026-07-18	BionicLoop engineering	SRS-OUTAGE-006 amended after independent safety review: gap-replay shares use 0.01 U schedule-weighted largest-remainder allocation so booked and measured totals match; the persisted plan carries credit context across termination; and schedule weights use the engaged event's recorded profile timezone with device-zone fallback.
1.44	2026-07-18	BionicLoop engineering	SRS-MEAL-002 amended (sponsor field report 2026-07-18): meal announce is additionally blocked while the primary algorithm is forced-open (backup basal running during a released CGM outage) - a meal bolus cannot deliver in that state; new <code>backupBasalRunning</code> unavailable reason, self-clearing on BG entry or sensor recovery. Replaces the prior behavior where announce was allowed and the delivery modal spun indefinitely.
1.45	2026-07-18	BionicLoop engineering	Documentation truth-sync: SRS-STATE-004 no longer claims that active issued-dose reconciliation holds persist. The writer-less hold state and its guards were removed in <code>c42f2a6</code> ; pending issued-dose attribution, consumed-delivery deduplication, discrepancy evidence, and reconciliation-failure lifecycle records remain the implemented controls. SRS-SEC-002 now includes the implemented protection/backup-exclusion requirement for app-enumerated Algo2015 diagnostic artifacts.

Version	Date	Author	Summary of Changes
1.46	2026-07-19	BionicLoop engineering	Pending manual-BG and alert-precedence hardening: SRS-BG-004/SRS-MEAL-002 allow meal announce during forced-open only when a valid persisted fingerstick is targeted to the exact prospective meal step, while retaining every pump/reconciliation gate; SRS-ALERT-011 gives active pump signal loss precedence over same-severity check-BG prompts without overriding safety-critical severity.
1.47	2026-07-20	BionicLoop engineering	Manual-BG confirmation and request-emphasis hardening: SRS-BG-001 requires a spatially separate exact-value review confirmation; SRS-OUTAGE-003 extends Home BG-entry emphasis to direct due-soon and overdue fingerstick requests.
1.48	2026-07-20	BionicLoop engineering	SRS-PUMP-010 fresh-completion upgrade rule (field incident on 2026-07-20, sponsor-approved): the relaunch persisted-evidence deferral/suppression no longer freezes a non-final in-progress or assumed delivered figure when a fresh same-pod settled status at/after expected physical completion reports <code>bolusNotDelivered = 0</code> — the adapter reconstructs and surfaces the pump-observed completed delivery, and matching non-assumed evidence below the requested amount is replaced by the strictly-higher same-pod post-completion pump observation before consumption. Canceled/interrupted partials (nonzero register), different/unproven pods, active-delivery, and pre-completion early-read shapes are explicitly excluded; no delivered figure is ever raised on assumption alone.
1.49	2026-07-20	BionicLoop engineering	SRS-MEAL-006 amended (sponsor-approved field report 2026-07-20): when the open composer was unlocked by the pending-fingerstick carve-out and a loop step consumes that BG before submit while the algorithm stays forced-open, the inline blocked messaging shall state the BG was already used by a dosing step and request another fingerstick or sensor recovery — not the generic backup-basal copy; selections stay preserved and the existing in-place re-arm / inline-clear recovery resumes the flow. The re-closed-loop sub-shape (BG-consuming step returns to closed loop) continues to heal through the existing stale-observed-step in-place re-arm.
1.50	2026-07-20	BionicLoop engineering	SRS-MEAL-006 consumed-carve-out clause corrected (sponsor correction 2026-07-20 of the 1.49 amendment, which mis-modeled the clinical behavior): an accepted fingerstick BG restarts the full CGM-dropout tolerance window and its consuming step publishes closed-loop state, so forced-open availability after the pending BG cleared but before the consuming step publishes is transient continuity — the composer holds armed on a passive waiting notice (never a demand for another fingerstick within the BG validity window) until that step's publication re-arms it, and the same hold applies to a backup-basal-blocked submit during the in-flight

Version	Date	Author	Summary of Changes
			window. Generic backup-basal messaging applies only when forced-open persists after the consuming step published (pending BG expired unconsumed or not accepted).
1.51	2026-07-21	BionicLoop engineering	SRS-LOG-009 amended (sponsor request 2026-07-20, steady-cadence fallback schedule visibility): <code>schedule_checked_unchanged</code> events now carry the programmed four-entry schedule plus profile coverage metadata (omitting only the bulk 288-rate array); <code>arm/refresh/unchanged</code> events add per-six-hour-segment observed-slot counts (<code>profile_observed_slot_counts_by_bucket</code>); Recent Dose Steps schedule rows show the four segment rates, an armed/updated/unchanged outcome label, and per-segment observed-vs-imputed provenance (<50% observed marks imputed).
1.52	2026-07-21	BionicLoop engineering	SRS-PUMP-010 fresh-completion evidence rule corrected after review against the 2026-07-08 canceled-bolus field incident: <code>bolusNotDelivered = 0</code> after relaunch is not completion proof for persisted partial evidence. A partial may be upgraded only when durable provenance identifies it as capped while delivery was still active; canceled and legacy-unclassified partial evidence remains authoritative. Assumed-full evidence may surface a matching fresh pump observation without increasing the conservative amount.
1.53	2026-07-21	BionicLoop engineering	SRS-MEAL-002 prospective forced-open guard: meal announce is blocked before dispatch when its borrowed/due step would be the 13th glucose-blind step and enter Algo2015 forced-open mode, even if the prior persisted output was still closed-loop. The same policy runs in UI preflight and in the coordinator before meal persistence; fresh CGM or a valid exact-step fingerstick permits the step. Blocked copy confirms the meal was not announced and no meal insulin was delivered.
1.54	2026-07-22	BionicLoop engineering	SRS-MEAL-002 presentation precedence corrected: when the active connected pod reports an in-progress delivery, meal preflight reports the pump as busy delivering even if issued-dose attribution is also pending; reconnect guidance remains for idle unresolved attribution.
1.55	2026-07-22	BionicLoop engineering	SRS-MEAL-006 open-composer recovery corrected for intermediate runtime publication: pending-BG consumption remains in flight until the exact step's qualifying-glucose marker publishes, even when a pre-command checkpoint already exposes that step as executed. Qualifying-glucose and live pump-status changes now trigger revalidation so recovered availability re-arms the composer and connected active delivery

Version	Date	Author	Summary of Changes
			promptly replaces stale backup-basal guidance with pump-busy guidance.
1.56	2026-07-24	BionicLoop engineering	Added SRS-CLIN-016/017 and SRS-LOG-013 for the Pregnancy Temporary Target, explicit Standard-to-Pregnancy 90% meal draft default, and lifecycle/per-step app-to-Scout monitoring contract.
1.57	2026-07-24	BionicLoop engineering	SRS-CLIN-014 reset telemetry ordering and reset-boundary concurrency aligned to the reviewed implementation: ordinary prior-participant uploads must quiesce before durable outbox erasure; inability to quiesce or persist the erase fails closed; the best-effort discard marker is initiated only after erasure; live pod/session guards are rechecked after awaits; algorithm arming is blocked while reset owns the transaction; and all outgoing Temporary Target state, including pending lifecycle records, is removed by the full participant wipe.
1.58	2026-07-24	BionicLoop engineering	Clarified SRS-LOG-013 reset telemetry boundaries: same-participant/session reset emits the lifecycle record, while New Participant Reset erases the outgoing participant identity and pending lifecycle state without emitting a new outgoing-participant event.
1.59	2026-07-30	BionicLoop engineering	Extended SRS-CLIN-016 so the combined Exercise and Insulin Delivery screen directly owns the existing Pod suspend/resume control, the duplicate control is absent from normal Pod Settings, reminder duration remains reminder-only, and suspend-alert navigation reaches the relocated control.
1.60	2026-07-30	BionicLoop engineering	Clarified SRS-SEC-009 login recovery: signed-out active-therapy Home keeps a persistent top-level Log In control independent of alert ordering; the active alert, Alert Center, Account & Session, and alert notification all provide a direct route to login without stopping the local algorithm session.
1.61	2026-07-30	BionicLoop engineering	Extended SRS-CLIN-016 with Exercise and Insulin Delivery placement above User, app-standard Temporary Target selection controls, removal of the unrelated reset-location text, and explicit discard confirmation when a target or duration draft would otherwise be lost before review/activation.
1.62	2026-07-30	BionicLoop engineering	Extended SRS-UI-002 so the Home Manual BG and Meal Announcement actions remain visible and actionable while status, alert, and chart content scrolls; action eligibility and handlers remain unchanged.
1.63	2026-07-30	BionicLoop engineering	Amended SRS-LOG-012 so permanent telemetry rejection remains durable, structured local evidence and a BionicScout sequence gap without a participant-facing generic upload alert; actionable subject-identity conflicts retain their specific alert,

Version	Date	Author	Summary of Changes
			rate limits remain retryable, and a one-time launch migration removes earlier-build generic alerts and notifications.
1.64	2026-07-30	BionicLoop engineering	Extended SRS-CLIN-016 so an ended suspension reminder offers a direct Resume Insulin command, remains active until pump-confirmed recovery, and escalates to safety-critical with a fresh notification when insulin remains suspended for 15 minutes.
1.65	2026-07-30	BionicLoop engineering	Tightened SRS-MEAL-002/SRS-PUMP-010/SRS-UI-002 for suspension: meal entry requires a confirmed idle Pod before algorithm execution, pump-confirmed resume clears the block without waiting for a loop step, definitively blocked bolus metadata cannot be reconstructed as delivery feedback, and Home displays persistent paused-delivery status.
1.66	2026-07-30	BionicLoop engineering	Added explicit and persistent G7 replacement acquisition, a staleness-never-releases identity control, ten-minute in-app acquisition guidance, and durably deduplicated adoption/stall telemetry requirements (SRS-CGM-006, SRS-ALERT-020, SRS-LOG-014).
1.67	2026-07-31	BionicLoop engineering	Tightened SRS-ALERT-014 blocker diagnosis so participant copy follows the latest unresolved runtime outcome or authoritative pump condition, represents suspension and fallback recovery explicitly, and cannot remain CGM-specific after contradictory evidence or a successful live step; clarified SRS-CLIN-014 reset isolation for asynchronous device-state persistence and telemetry completion.
1.68	2026-08-06	BionicLoop engineering	Extended SRS-LOG-001 so Recent Dose Steps identifies valid manual/fingerstick BG input, shows the exact value used, and preserves any concurrent CGM evidence.
1.69	2026-08-11	BionicLoop engineering	Added SRS-ALG-009 after field telemetry exposed meal-step pump commands that omitted the concurrent basal component and therefore failed the Algo2015 delivered-insulin reconciliation gate.
1.70	2026-08-11	BionicLoop engineering	Corrected SRS-CLIN-002/013 document drift: initial clinical configuration is available without an unlock only while no usable subject configuration exists; every later Clinical Settings entry/edit remains subject-scoped offline-unlock gated.
1.71	2026-08-13	BionicLoop engineering	Hardened SRS-LOG-012 for incident-scale offline telemetry: update-safe spool migration, incremental protected persistence, retained-evidence-first recovery, coalesced chatter-drop reporting, transition-only notification telemetry, and passive backlog visibility.
1.72	2026-08-17	BionicLoop engineering	Strengthened SRS-LOG-001/012 with complete active-session protected step evidence, an uncapped retained-clinical SQLite

Version	Date	Author	Summary of Changes
			ledger, durable-enqueue/network-drain separation, retry liveness, and backward-compatible bounded batch ingest.
1.73	2026-08-20	BionicLoop engineering	Added SRS-BG-013 so Home distinguishes an accepted pending fingerstick BG from completed algorithm-use evidence without moving or duplicating the chart marker.
1.74	2026-08-20	BionicLoop engineering	Added SRS-LOG-015 for clinical-gated, session-scoped export of byte-faithful legacy Algo2015 artifacts using epoch grouping, bounded snapshots, partial-set disclosure, and protected temporary ZIP staging.
1.75	2026-08-20	BionicLoop engineering	Replaced the competing Settings export presentations with the established Recent Dose Steps share action and one protected ZIP containing the complete step CSV plus the matching/current legacy Algo2015 session artifacts.
1.76	2026-08-20	BionicLoop engineering	Extended SRS-ALERT-014 so fresh idle Pod status after a pump-status-unavailable attempt updates the existing interruption alert to communication-restored/waiting-for-glucose guidance without triggering algorithm work; stale or non-idle status cannot claim recovery.
1.77	2026-08-20	BionicLoop engineering	Extended SRS-UI-008 so the Home CGM chart renders discrete glucose points without a connector line or interpolated area fill while preserving point color, time placement, and scrub selection.
1.78	2026-08-21	BionicLoop engineering	Bound the controlled specification to the immutable 2026-08-21 software freeze and clarified the pending approval boundary. No requirement or product behavior changed.
1.79	2026-08-27	BionicLoop engineering	Replaced internal development labels and test-device shorthand with reviewer-neutral descriptions and updated the approval role. No requirement or product behavior changed.
1.80	2026-08-27	BionicLoop engineering	Aligned SRS-SEC-001 with the clinically approved supportive Scout boundary and official insulin-delivery source, recorded the confirmed participant-weight range, and corrected reset-drain wording. No product behavior changed.
1.81	2026-08-28	BionicLoop engineering	Added requirement-control methodology, applicability and safety-significance classification, controlled threshold definitions, addressable compound-clause ranges, and complete deferred/retired traceability disposition. Replaced implicit timing terms with frozen-baseline values. No product requirement or behavior changed.
1.82	2026-09-03	Software Developer	Completed the reviewer-style consistency audit: made frozen thresholds explicit, removed internal working-record citations from submission-facing text, clarified deferred security scope,

Version	Date	Author	Summary of Changes
			and aligned controlled-source traceability. No product behavior or frozen-baseline requirement changed.

Historical interim revision identifiers such as 1.111 and 1.151 preserve the original controlled record and are not a chronological renumbering error.

Requirement Format

Each requirement uses a stable parent ID. A parent containing one normative `shall` or `shall not` occurrence is an atomic requirement. A parent containing multiple normative occurrences is a **requirement family** retained to preserve the frozen design and evidence links.

Within a requirement family, each normative occurrence has an addressable child clause ID formed by appending a two-digit sequence in textual order, beginning with `.01`. For example, the first and second normative occurrences in `SRS-PUMP-010` are `SRS-PUMP-010.01` and `SRS-PUMP-010.02`. A child clause is read with the subject, conditions, and definitions in its governing sentence. The sequence is controlled for this revision and shall not be reordered or reused without a documented traceability impact assessment.

RTM mapping to a parent applies to all of its child clauses. Parent-level verification or approval may be closed only after every applicable child clause has objective evidence or a documented disposition. Explanatory text under `Team review notes`, `evolution notes`, and revision history is informative and not an additional requirement.

Applicability

- **Active:** normative for the frozen baseline.
- **Deferred:** excluded from the frozen baseline; implementation or working evidence does not make it an accepted requirement without explicit scope re-entry.
- **Retired:** historical identifier retained for traceability and prohibited from reuse.

All requirements are Active except `SRS-BG-008` and `SRS-SEC-003..009`, which are Deferred, and `SRS-LOG-010`, which is Retired.

Safety Significance

- **C1 - direct therapy control:** failure could directly affect insulin calculation, command authorization, delivery accounting, fallback delivery, device identity, or safety-critical clinical configuration.
- **C2 - supporting safety control:** evidence integrity, participant communication, cybersecurity, or operational support that does not itself authorize insulin delivery.
- **D - deferred or retired:** not part of the frozen-baseline claim.

Requirement set	Applicability	Safety significance
SRS-RUN-001..007	Active	C1
SRS-CGM-001..006	Active	C1
SRS-BG-001..007, SRS-BG-009..012	Active	C1
SRS-BG-013	Active	C2
SRS-BG-008	Deferred	D
SRS-PUMP-001..011	Active	C1
SRS-MEAL-001..012	Active	C1
SRS-OUTAGE-001..006	Active	C1
SRS-STATE-001..005	Active	C1
SRS-ALG-001..009	Active	C1
SRS-LOG-001..009, SRS-LOG-011..015	Active	C2
SRS-LOG-010	Retired	D
SRS-UI-002, SRS-UI-004, SRS-UI-005, SRS-UI-007	Active	C1
SRS-UI-001, SRS-UI-003, SRS-UI-006, SRS-UI-008, SRS-UI-009	Active	C2
SRS-CLIN-001..017, SRS-VAL-001	Active	C1
SRS-ALERT-019	Active	C1
SRS-ALERT-001..018, SRS-ALERT-020	Active	C2
SRS-SEC-001..002	Active	C2
SRS-SEC-003..009	Deferred	D

The C1 set is the reviewer-priority critical-requirement index for this SRS. The Risk Analysis and RTM remain authoritative for hazard linkage and residual risk; this classification does not replace those records.

Controlled Terms and Thresholds

Term	Frozen-baseline definition
Approved CGM freshness limit	Accepted CGM receipt age no greater than 5 minutes for algorithm-use and reconnect-fallback eligibility.
Approved interruption threshold	More than 15 minutes without a successful algorithm step while the session is armed.

Term	Frozen-baseline definition
Algo2015 CGM-dropout tolerance	60 minutes, represented by 12 consecutive five-minute blind steps after the last accepted CGM or fingerstick BG input; forced-open behavior begins on the next blind step.
Approved no-active-Pod/signal-loss debounce	5 minutes; display and alert debounce do not alter pump-command or dosing safety state.
Approved borrow window	At most 2 future five-minute slots, as constrained by SRS-MEAL-001.
Approved renewal window (fallback mask)	The final 20 minutes of the active 30-minute 0.0 U/hr mask.
Issued-dose request-unit match tolerance	Absolute requested-unit difference no greater than 0.025 U; delivered units must remain within 0...requested with a numerical tolerance of 0.0001 U.
Trustworthy G7 reading, for SRS-ALERT-016	A reading with the G7 reliable-glucose flag, a glucose value, and an observation age no greater than 11 minutes. This alert criterion does not broaden the separate 5-minute algorithm-input freshness limit.
Fresh pump status	A status obtained by the current required refresh operation rather than restored or cached status.
Settled pump status	A fresh pump delivery state of idle or suspended, not delivering or unknown.
Valid fallback candidate	A candidate derived from the current or latest successful algorithm step for the current subject/session and usable by the fallback schedule/rate selection defined in SRS-PUMP-006 and SDD-PUMP-001.
Credible pump-delivery evidence	Evidence satisfying the applicable pod-identity, request-step, measurement-window, delivered-range, schedule-compatibility, and partitioning controls in SRS-PUMP-009, SRS-PUMP-010, and SDD-PUMP-001.
Normal execution safety gates	The active-session, due-step, device-identity, fresh pump-status, delivery-state, suspension, fallback-recovery, and issued-dose reconciliation gates applicable to the requested wake path. The controlling requirements are SRS-RUN-002...006, SRS-PUMP-001...011, and SRS-STATE-001...005.
Protected local evidence	Exported CSV and temporary recovery-export staging use complete file protection. The telemetry ledger, current-session step archive, and enumerated live Algo2015 artifacts

Term	Frozen-baseline definition
	use complete-until-first-user-authentication protection. All are backup-excluded; see SRS-SEC-001..002 and SDD-LOG-001.

Compound Requirement Family Index

The following high-density families are called out because their clauses span multiple failure and recovery paths. Clause counts are the normative shall / shall not occurrences in this revision.

Requirement family	Child clause range	Primary verification mapping
SRS-RUN-002	.01..12	TV-RUN-001, TV-RUN-002, TV-RUN-005, TV-RUN-006, TV-MEAL-001, TV-MEAL-011, TV-SIM-001
SRS-RUN-006	.01..27	TV-RUN-008, TV-SIM-POD-001
SRS-PUMP-009	.01..13	TV-PUMP-008, TV-SIM-POD-003, TV-SIM-POD-004
SRS-PUMP-010	.01..34	TV-PUMP-009, TV-MEAL-013, TV-SIM-POD-004
SRS-MEAL-002	.01..16	TV-MEAL-002, TV-MEAL-005, TV-SIM-004
SRS-MEAL-006	.01..15	TV-MEAL-007
SRS-OUTAGE-006	.01..14	TV-OUTAGE-002, TV-PUMP-008, TV-RUN-008
SRS-LOG-009	.01..13	TV-LOG-009, TV-SIM-POD-002..004
SRS-LOG-012	.01..21	TV-LOG-012
SRS-CLIN-002	.01..15	TV-CLIN-001
SRS-CLIN-014	.01..14	TV-CLIN-015
SRS-CLIN-016	.01..22	TV-CLIN-017, TV-LOG-013, TV-ALERT-009
SRS-ALERT-014	.01..10	TV-ALERT-013
SRS-SEC-001	.01..10	TV-SEC-001, TV-LOG-015
SRS-LOG-015	.01..12	TV-LOG-015

The complete family-range register below makes every compound parent directly auditable. Parents not listed contain one normative occurrence and are atomic at the parent ID; SRS-LOG-010 is the retired zero-clause exception.

- **Runtime:** SRS-RUN-002 (.01..12), SRS-RUN-005 (.01..03), SRS-RUN-006 (.01..27), SRS-RUN-007 (.01..03).

- **CGM and BG:** SRS-CGM-006(.01..05), SRS-BG-001(.01..03), SRS-BG-002(.01..02), SRS-BG-004(.01..03), SRS-BG-005(.01..02), SRS-BG-006(.01..03), SRS-BG-013(.01..04).
- **Pump:** SRS-PUMP-006(.01..07), SRS-PUMP-007(.01..02), SRS-PUMP-008(.01..08), SRS-PUMP-009(.01..13), SRS-PUMP-010(.01..34), SRS-PUMP-011(.01..03).
- **Meal:** SRS-MEAL-002(.01..16), SRS-MEAL-006(.01..15), SRS-MEAL-007(.01..04), SRS-MEAL-008(.01..02), SRS-MEAL-009(.01..04), SRS-MEAL-010(.01..02), SRS-MEAL-012(.01..05).
- **Outage, state, and algorithm:** SRS-OUTAGE-001(.01..02), SRS-OUTAGE-002(.01..02), SRS-OUTAGE-003(.01..04), SRS-OUTAGE-004(.01..03), SRS-OUTAGE-006(.01..14), SRS-STATE-004(.01..05), SRS-STATE-005(.01..02), SRS-ALG-007(.01..02), SRS-ALG-008(.01..02), SRS-ALG-009(.01..02).
- **Logging:** SRS-LOG-001(.01..09), SRS-LOG-007(.01..02), SRS-LOG-009(.01..13), SRS-LOG-011(.01..06), SRS-LOG-012(.01..21), SRS-LOG-013(.01..04), SRS-LOG-014(.01..05), SRS-LOG-015(.01..12).
- **UI and clinical configuration:** SRS-UI-002(.01..04), SRS-UI-005(.01..02), SRS-UI-006(.01..03), SRS-UI-008(.01..03), SRS-CLIN-002(.01..15), SRS-CLIN-011(.01..02), SRS-CLIN-013(.01..08), SRS-CLIN-014(.01..14), SRS-CLIN-015(.01..05), SRS-CLIN-016(.01..22), SRS-CLIN-017(.01..02).
- **Alerts and security:** SRS-ALERT-005(.01..02), SRS-ALERT-011(.01..07), SRS-ALERT-012(.01..02), SRS-ALERT-014(.01..10), SRS-ALERT-015(.01..05), SRS-ALERT-016(.01..06), SRS-ALERT-017(.01..02), SRS-ALERT-018(.01..04), SRS-ALERT-019(.01..07), SRS-ALERT-020(.01..04), SRS-SEC-001(.01..10), SRS-SEC-002(.01..03), SRS-SEC-009(.01..04).

System Boundary and Operating Context

This SRS is read with the controlled IDE Software Device Description and Software Version and Configuration Identification. Those records identify the exact app, iOS/toolchain configuration, local packages, algorithm source, dependency resolution, and frozen source commit; the SRS does not duplicate configuration identifiers that are controlled there.

The software boundary comprises the BionicLoop iPhone application, BionicLoopCore runtime and safety policy, the two isolated Algo2015 controller instances, G7SensorKit/G7SensorKitUI, OmniBLE, LoopKit, and protected local state/evidence stores. External interfaces are Dexcom G7 and Omnipod DASH over Bluetooth Low Energy, iOS lifecycle/notification services, clinical-gated local export, and supportive BionicScout telemetry. Cloud availability is not an input to local dosing authorization.

Principal inputs are CGM and fingerstick glucose, pump status and delivery feedback, meal selections, clinical configuration, time/session state, and explicit participant actions. Principal outputs are algorithm records, allowed pump commands, participant status/alerts, protected local

evidence, and supportive telemetry. Ranges, limits, defaults, error handling, and recovery behavior are specified in the requirement families below.

Runtime and Cadence

- SRS–RUN–001: The runtime shall maintain 5-minute algorithm cadence anchored to first successful step execution time.
- SRS–RUN–002: The runtime shall compute expected step index from anchored schedule and skip duplicate step execution for the same index. Before resolving ordinary cadence or meal-announce step selection, runtime shall reconcile persisted cadence state with the algorithm-reported next step; if the algorithm has already consumed the current due step while the persisted runtime state lags, runtime shall advance persisted `lastExecutedStep` to the algorithm-implied last executed step without re-anchoring the schedule, so live `executedStep`, pump request-step attribution, and algorithm input `stepTime` remain aligned. Loop work passes shall be single-flight: at most one load-maintain-persist-execute pass may be in flight at a time, concurrent wake triggers shall queue FIFO, and each queued pass shall load persisted state when it actually starts. If the algorithm-reported next step is ahead of the runtime step base beyond the one-step reconcile, runtime shall adopt the algorithm counter as the step base without re-anchoring the schedule, skip until wall clock reaches the adopted counter while fallback basal coverage remains in force, publish an informational auto-clearing re-sync note stating the resume window, and resume automatically; runtime shall not halt dosing or demand a manual algorithm reset on step mismatch. Pre-execution state persistence shall be monotonic within a session anchor: a pass shall not persist a `lastExecutedStep` older than the currently persisted value for the same `firstSuccessfulStepAt` anchor and shall instead abort benignly as a superseded pass; session resets with a different or cleared anchor may regress freely. Algorithm step-drift blocking shall therefore only ever be evaluated against state at least as fresh as the last completed pass.
- SRS–RUN–003: The runtime shall execute loop work only on the approved configured wake causes: `cgmUpdate`, `bgCheck`, `mealAnnounce`, and guarded `pumpReconnect` fallback paths.
- SRS–RUN–004: After the first successful anchored step exists, the runtime shall permit a guarded pump-reconnect fallback wake to execute the current due step when no accepted CGM receipt newer than the 5-minute CGM freshness limit exists and normal execution safety gates pass.
- SRS–RUN–005: Ordinary guarded pump-reconnect fallback for CGM absence shall not execute step 0, shall not re-anchor cadence, and shall not replay multiple missed slots; each ordinary reconnect-triggered execution may service at most the current due step. This row does not describe masked offline-fallback recovery after a persisted reconciliation-required state.

- SRS-RUN-006:** Masked offline-fallback shall not arm a new fallback schedule or refresh the programmed fallback schedule unless the loop is armed, a current algorithm session identifier exists, and a valid fallback candidate is available from either the current just-computed algorithm step or a prior successful step. When a current step first produces a valid fallback candidate and no fallback is armed, runtime shall attempt the first fallback arm after primary/secondary algorithm calculation and before applying that same step's pump command, so step 0 and other first-candidate steps cannot let a bolus outrun fallback arming. Pre-execution masked-fallback arm, mask-renewal, and schedule-refresh maintenance shall require freshly refreshed idle pump status; unavailable, unknown, delivering, or refresh-failure status shall defer that maintenance and record the defer reason without mutating pump state. An already-masked fallback may renew its existing 0.0 U/hr mask before the next step executes once the runtime is inside the approved renewal window, but only while the current mask remains active and pump maintenance safety gates pass. If runtime later detects offline mask expiry (`maskExpiredOffline`) or any remask failure leaves the fallback schedule potentially exposed after possible schedule mutation or mask replacement, it shall persist a masked-fallback reconciliation-required recovery state, suppress ordinary loop `doWork` execution, and keep closed-loop stepping blocked until reconnect recovery either resolves by connected restore/disarm or is otherwise explicitly cleared; it shall not silently auto-resume authoritative closed loop in the same session immediately after detecting that exposure state. For the accepted baseline, if reconnect restore succeeds while the loop remained armed and basal-only reconnect evidence is `confirmed` or `corrected`, runtime shall preserve the existing session/cadence anchor, bound replay to missed steps before the resumed current due step whose delivery interval overlaps confirmed fallback-active time, and replay those primary and secondary/safety algorithm steps with unavailable CGM input (`CGM=-1`) and per-step pump-delivery input allocated from credible pump-reported delivered insulin using the persisted programmed fallback schedule as the per-step weighting model. Each fallback-only replay row after the first shall report `unitsRequested` as the algorithm's own issued ask from the preceding replay row: the Algo2015 reconcile gate only credits a reported delivery whose requested amount matches its own remembered ask within one mechanical step (0.01 U), so echoing the actual ask credits pod-measured shares at any fallback rate. When pre-fallback issued-dose replay rows precede the fallback window, the first fallback replay row shall echo the last preceding issued-dose replay row's ask across the seam; only a classic fallback recovery with no preceding replay rows keeps the allocated-share proxy on its first row (a released-outage pod-away gap replay instead echoes the persisted pre-gap ask carried in the plan's credit context, per SRS-OUTAGE-006). Disconnected missed steps before offline fallback activation shall remain missing and shall not be replayed as synthetic 0 U rows, except when a resolved issued dose requires pre-fallback attribution replay: the dead-gap window from the dose's first eligible attribution step to the step before fallback activation shall then be replayed as a pure issued-dose confirm row - whose pump tuple echoes the issued ask exactly and reports the reconciled delivered amount, so the reconcile gate cannot be broken by a fallback share riding on the requested side - followed by zero-delivery echo sentinel rows that retract each preceding offline ask (the pod was idle; these rows claim no fallback exposure).

When the fallback window begins on the first eligible attribution step itself (no dead-gap room), the resolved issued dose shall merge into the first fallback replay row with `unitsRequested` echoing the issued ask exactly and the fallback share carried only in `unitsDelivered`; the algorithm credits that share as an over-delivery adjustment at its last-run buffer position (the request step), an accepted clinical policy bounded to one slot's fallback share booked early. The replayed steps shall not issue pump commands. The resumed live step shall then execute with actual refreshed CGM/pump status and shall not receive a duplicate aggregate recovered-delivery input. If reconnect evidence is ambiguous, runtime may resume the existing session on the current due 5-minute step without fallback delivery reconciliation, but that path shall be marked as an unreconciled resume rather than authoritative reconciliation, and when nonzero fallback insulin may have been delivered (pump-reported delta or modeled exposure greater than zero) runtime shall raise a safety-critical operator alert rather than resume silently. If connected restore/disarm cannot complete because the previous pod is known inactive/no-active, retired, or past hard service-stop while masked-fallback recovery remains pending, runtime may clear the recovery state without creating a new algorithm session by recording modeled fallback exposure as `assumed_delivered_per_clinical_policy`, queueing the same bounded no-command replay plan, and labeling the recovery as assumed rather than pump-confirmed. While a masked-fallback reconciliation-required recovery state remains pending, runtime shall keep reconnect recovery attempts live without requiring user interaction: any ordinary wake blocked by the pending state (including CGM-driven `doWork`) shall re-schedule a reconnect recovery attempt, and a recovery trigger arriving while an attempt is already in flight shall be retained and re-evaluated after that attempt completes rather than dropped. A queued fallback replay plan shall carry the reconciliation status of the recovery that authorized it, and replay authorization shall be judged by that stamp - a later fallback lifecycle event (such as the post-recovery re-arm, which is always pending) shall not mask an authorized plan into disposal; plans persisted before the stamp existed retain the legacy latest-event check. Pump-counter replay evidence shall be evaluated against the interval the measurement actually covers: the credited exposure window is bounded to the pod-total measurement time, recovery-completion latency after that measurement bounds rather than disqualifies the evidence, and the uncovered post-measurement tail is not replayed. After a successful connected restore, runtime shall prefer pod totals re-read after the restore (totals-only overlay preserving pre-disarm schedule and temp-basal evidence) so fallback delivered during the disarm itself is measured and credited. A same-pod pump-counter delta that does not exceed modeled fallback exposure plus tolerance shall reconcile as corrected pump-anchored replay even when the delivery baseline exceeds the freshness limit; the unreconciled resume is reserved for genuinely unusable evidence (no delta with mismatched schedule evidence, over-tolerance delta, or unproven pod continuity).

- **SRS-RUN-007:** The runtime shall never terminate an algorithm session or block dosing pending manual user intervention as an automatic failure response. Automatic failure handling shall degrade to fallback basal coverage and self-healing realignment; algorithm session reset shall remain available only as an explicit user-initiated action.

Team review notes:

- SRS–RUN–003 reflects the currently implemented wake-cause policy.
- Guarded reconnect fallback is implemented as a secondary wake path from pump-status recovery, not raw BLE connect.
- Current reconnect-fallback gate is `accepted CGM receipt age > 5 minutes`, matching the present CGM freshness limit for algorithm-use eligibility.

CGM Input Handling

- SRS–CGM–001: The runtime shall map CGM values outside `39...401 mg/dL` to unavailable input (`-1`) before algorithm invocation.
- SRS–CGM–002: Step 0 shall require a fresh in-range CGM input.
- SRS–CGM–003: Step `>0` shall be allowed to run with unavailable CGM input (`-1`) when fresh in-range CGM is not available.
- SRS–CGM–004: The UI shall show loop reason/state when step execution is blocked for missing fresh CGM at step 0.
- SRS–CGM–005: When the loop is armed and no successful algorithm step has completed for more than `15 minutes`, the app shall detect that loop stepping is stalled and expose that condition to alert/status logic.
- SRS–CGM–006: Dexcom G7 Settings shall always provide an explicit `Scan for new sensor` action independent of routine BLE scanning for the currently bound sensor. Invoking replacement acquisition shall durably record its start time, prior sensor identity, and trigger before clearing the old binding; relaunch shall preserve that acquisition episode; successful authenticated sensor adoption shall clear it. CGM reading age or connection staleness alone shall never clear the bound sensor identity or initiate replacement adoption.

Team review notes:

- SRS–CGM–005 is implemented with a 15-minute threshold based on the most recent successful step time; before the first successful step exists, the threshold is measured from session start / first-step wait state.
- The interruption condition is recomputed on foreground from persisted runtime/session state and is distinct from the protocol Dexcom Bluetooth interruption alert (`> 2 hours`).

Fingerstick BG Input Handling

- SRS–BG–001: The app shall provide a manual BG entry flow with explicit `Review`, `exact-value Use`, `Change`, and `Cancel` actions; the exact-value confirmation shall appear on a spatially separate review surface so repeated taps at the entry action cannot submit without explicit confirmation. The app shall enforce accepted input range `20...600 mg/dL`, capture a submission timestamp at user confirmation time, and show user-facing guidance that manual BG is used for algorithm dosing decisions and does not calibrate CGM.

- SRS-BG-002: Manual BG submit shall trigger a dedicated runtime wake cause (`bgCheck`) with no future-slot borrowing and shall create/update a single pending BG candidate for the next eligible algorithm step.
- SRS-BG-003: If submit occurs after the current due step has already executed, the pending BG candidate shall roll forward to the immediate next step (not discarded immediately).
- SRS-BG-004: During `bgCheck`, manual BG shall map to algorithm BG input (`BGval`) while CGM input mapping policy remains independent (CGM may be unavailable -1). If meal announce executes the exact step targeted by a valid pending manual BG, that single step shall carry both the manual BG and meal inputs to the primary and safety algorithms, and shall consume the pending BG once.
- SRS-BG-005: BG submission shall not override pump command-application safety gates; when pump status is unavailable/unknown, runtime may execute algorithm step but shall block command application.
- SRS-BG-006: Runtime shall bound manual BG staleness through step-scoped candidate validity: a submitted BG is valid only for the immediate next eligible algorithm step and shall be discarded unconsumed once that step passes (see SRS-BG-009), so the BG age at algorithm consumption cannot exceed one cadence interval plus the SRS-MEAL-001 borrow window of at most 2 future five-minute slots. Rejected or discarded submissions shall produce an explicit user-visible reason.
- SRS-BG-007: Telemetry shall record BG source (`manualBG`), submitted value, submit timestamp, and execution result for each `bgCheck` attempt.
- SRS-BG-008: Step-0 BG rescue is deferred from the current software baseline. If a future approved baseline enables step 0 BG rescue, it shall permit execution only when fresh in-range CGM is unavailable and a valid fresh manual BG is present.
- SRS-BG-009: Pending BG candidate shall be valid for one step only and must be discarded if not consumed on that immediate next step.
- SRS-BG-010: If a new manual BG is submitted while a pending BG candidate exists and has not yet been consumed, the newer submission shall replace the older candidate.
- SRS-BG-011: Manual BG submit shall not execute `bgCheck` when loop runtime is disarmed (`Algorithm Off` state) or when masked-fallback reconciliation is pending.
- SRS-BG-012: Until step-0 BG rescue is explicitly approved, manual BG submit shall be rejected before the first successful anchored step exists (`firstSuccessfulStepAt == nil`).
- SRS-BG-013: After a manual BG candidate is accepted and persisted, Home shall immediately render an unfilled red blood-drop chart marker at its submission time. The marker shall become filled only when a completed algorithm step records that BG as consumed. Pending and consumed evidence for the same target step shall render as one marker at the original submission time; rejected, replaced, expired, or otherwise discarded unconsumed candidates shall not render as consumed.

Team review notes:

- SRS-BG-008 is deferred from the current software baseline; current accepted behavior remains CGM-only at step 0 via SRS-BG-012.
- Manual BG accepted range is currently set to 20 . . . 600 mg/dL.
- Staleness is enforced by step-scoped candidate validity. A separate wall-clock validity limit is not part of the frozen baseline and would require controlled requirements, risk, and verification review before inclusion.
- SRS-BG-011 is implemented in app runtime to prevent loop-off execution from manual BG UI.
- SRS-BG-012 is currently enforced in runtime; step-0 BG rescue remains disabled by default.

Pump Handling

- SRS-PUMP-001: On pump-status refresh failure or unknown state, runtime shall continue algorithm execution with unavailable pump input fields and block command application.
- SRS-PUMP-002: In closed-loop mode, manual bolus command paths shall not be exposed.
- SRS-PUMP-003: Dosing command handling shall respect DASH delivery constraints, including the supported 0.05 U command quantum.
- SRS-PUMP-004: Home pod status tile shall update to reflect actual connect/disconnect state transitions without requiring user entry into pump settings.
- SRS-PUMP-005: While pump delivery is in progress, Home and meal-availability pump state shall auto-refresh without requiring navigation into pump settings.
- SRS-PUMP-006: For the masked offline-fallback path, the app shall maintain a connected 0.0 U/hr / 30 minute temp-basal mask over the programmed fallback basal schedule while communications remain healthy. The programmed fallback schedule shall be derived from the secondary/safety algorithm track's persisted q5 nominal-basal profile when available by averaging four local-time six-hour blocks (00:00–06:00, 06:00–12:00, 12:00–18:00, 18:00–24:00) and rounding each block to a supported DASH basal rate; if no profile can be generated yet, runtime may fall back to the latest rounded scalar safety-track nominal-basal candidate. Existing masked fallback shall enter renewal once the active mask reaches its final 20 minutes; pre-execution renewal and schedule refresh shall first verify fresh idle pump status, and if renewal is temporarily blocked but the current mask remains active, runtime shall defer maintenance and record the defer reason rather than treating the attempt as an immediate hard failure. Programmed fallback schedule refresh shall use the IDE baseline cadence of no more often than every 6 hours unless explicitly overridden for controlled hardware validation. If successful step execution stalls or deferred maintenance continues long enough for the active mask to expire, the pod may autonomously resume the programmed fallback schedule and runtime shall record that transition for later reconciliation rather than continuing ordinary authoritative loop stepping.
- SRS-PUMP-007: Masked fallback schedule arm, refresh, and disarm shall be treated as disruptive programmed-basal replacements and shall be blocked unless pump delivery state is known and safe for maintenance (no active bolus, no unknown basal-delivery state, and no nonzero temp basal).

- SRS-PUMP-008:** If an active algorithm session has no currently armed masked fallback and a valid rounded nominal fallback candidate is available from the current or most recent successful step, runtime shall program the fallback schedule before any later same-step pump command when the candidate was just computed, or in the next safe pre-execution maintenance pass with fresh idle pump status when the candidate was already persisted, and shall apply the connected 0.0 U/hr mask in the same maintenance sequence. The pre-pump-command first-arm attempt shall run only with refreshed pump status that is not unavailable and not unknown; otherwise runtime shall leave first-arm maintenance for a later safe wake and allow the current step's normal pump command path to classify the unavailable or unknown pump state under its own safety rules. The programmed schedule shall prefer the secondary/safety q5 nominal-basal profile-derived four-block schedule over the scalar candidate once at least one valid q5 profile observation exists. If that first arm attempt is cleanly blocked before any schedule mutation, runtime shall leave fallback explicitly unarmed and record the blocked arm as a failure requiring retry, not as ordinary deferred masked maintenance, while the current step's normal pump command may still be attempted under its own safety gates; runtime shall not immediately retry that first arm again in the same cycle after emitting a pre-command fallback event. If schedule programming may have succeeded but the immediate 0.0 U/hr mask fails, is blocked, or is uncertain, runtime shall persist reconciliation-required masked-fallback recovery context and block the current step's normal pump command.
- SRS-PUMP-009:** Session reset/disarm after masked fallback has been armed shall attempt to restore the original programmed basal schedule and shall record restore success or failure for operator review. If restore fails, the app shall preserve recoverable restore context and block the next algorithm start until masked-fallback recovery is resolved. When masked-fallback recovery remains pending across reconnect, app start, or foreground, the app may retry connected restore/disarm of the original programmed schedule; if that retry succeeds after offline mask expiry or any remask failure that may have left the fallback schedule exposed while the loop remained armed and basal-only reconnect evidence is confirmed, corrected, or assumed_delivered_per_clinical_policy, it shall preserve the existing session/cadence anchor, clear the blocked recovery state, queue fallback delivery reconciliation, and then permit same-session recovery. Reconciliation shall use pump-reported delivered-insulin delta when same physical pod continuity is proven and the connected delivery-counter baseline credibly spans the exposure window. If pump-reported delivery evidence is unavailable rather than contradictory, same physical pod continuity is proven, recovered schedule evidence matches the persisted fallback schedule, and no pending issued meal/correction dose requires pump-total partitioning, runtime may record persisted-schedule modeled fallback exposure as assumed delivered per clinical policy. Historical CGM is not required to authorize fallback delivery reconciliation; instead, confirmed, corrected, or assumed recovery shall replay only the bounded missed primary and secondary/safety algorithm steps whose delivery interval overlaps the fallback-active interval, with $\text{CGM}=-1$, per-step pump-delivery input allocated from the authoritative pump-reported delta when available or from the assumed modeled exposure when pump evidence is unavailable, explicit 0 U input for fallback-active replayed steps with

no allocation, and no catch-up pump commands before resuming the current due step. Pre-activation disconnected missed steps shall not be replayed. When credible pump-reported delta exists, that delta shall remain the authoritative total; schedule-derived exposure shall be used only as the deterministic weighting model, including actual overlap duration when replay slots cross fallback-schedule boundaries or midnight. If the persisted schedule is unavailable or the schedule weights sum to zero while the reconciled/assumed delivered amount is positive, replay shall be suppressed rather than guessed; if the authoritative pump-reported delta is 0 U, replay may still backfill the fallback-active missed step range with explicit 0 U inputs. If reconnect evidence is ambiguous, from a different/new pod, from unknown pod identity, has a delivery-counter/history discontinuity, has recovered schedule evidence that contradicts the persisted fallback schedule without compatible pump-delivery evidence, or cannot safely separate pending meal/correction insulin from fallback basal exposure, the app may instead permit same-session resume on the current due 5-minute slot without fallback delivery reconciliation, but it shall record that outcome as an unreconciled resume for operator review. Successful reconnect recovery shall also retain modeled fallback exposure and, when compatible same-pod pump-evidence baseline data are available, the pump-reported delivered-insulin delta for operator review. If the operator explicitly reset or turned the loop off before recovery completes, the successful restore shall keep the loop off. Exception: when the original pod is known inactive/no-active, retired, or past hard service-stop and connected pump-reported evidence is therefore unavailable, the app may use the persisted fallback schedule to record modeled fallback exposure as assumed delivered per clinical policy, replay the bounded fallback-active missed steps without pump commands, retain the assumed-vs-confirmed distinction in UI/telemetry, and preserve the existing algorithm session.

- **SRS-PUMP-010:** For any automatic algorithm-issued bolus command (including meal, correction, mixed, or basal micro-dose commands), runtime shall persist issued-dose attribution when the command is applied or its delivery outcome is uncertain. The attribution shall include request step, first eligible attribution step, requested units, request time, expected physical completion time, component split when available, flow ID when applicable, pod identity when available, and the algorithm-visible request step used for meal-learning exclusion when available. If the next due step arrives while that dose is still delivering, runtime shall skip the algorithm step rather than advancing with provisional delivery evidence. The accepted DASH liveness bound for this no-boundary-cancel policy is that supported automatic bolus sizes physically complete before the next 5-minute step under fresh connected pod status; communication loss, pod expiry, and no-active-pod paths shall report unavailable/unknown rather than active delivery and therefore leave the active-delivery skip path. If communication resumes after one or more missed steps and pump evidence matches the same request step, requested units within supported rounding tolerance, proven same pod identity, and delivered units within the credible range of 0 . . . requested units (within tolerance), runtime shall replay the missed primary and secondary/safety algorithm steps before the live resumed step, inject the reconciled delivered amount only at the first missed step after the request step, use unavailable CGM input (CGM=-1) for synthetic missed steps, and issue no catch-up pump commands. If persisted matching evidence exists and the live due step is exactly the first

eligible attribution step, runtime shall feed that evidence into the live step without creating synthetic replay rows. Pump status restoration after relaunch shall pass persisted matching issued-dose reconciliation evidence into the pump adapter; when such evidence exists for the current request step, the adapter shall defer to that evidence and shall not reconstruct a new `lastDelivery` from pod `bolusNotDelivered` counters. Persisted pump-last-delivery evidence may be restored as the adapter's last delivery, while assumed-delivered or otherwise non-pump evidence shall suppress reconstruction without being surfaced as pump-observed delivery. Exception (fresh-completion upgrade rule): when the current refresh is a fresh same-pod observation, with proven pod identity and settled (idle/suspended) delivery state, taken at or after the attributed dose's expected physical completion time, and the pod's `bolusNotDelivered` register reports zero insulin undelivered, the adapter shall reconstruct and surface the pump-observed completed delivery instead of deferring, so a non-final in-progress or assumed evidence figure cannot permanently freeze the displayed/recorded delivered amount below what the pod actually delivered; runtime shall likewise replace matching non-assumed issued-dose reconciliation evidence whose delivered figure is below the requested amount with a fresh same-pod settled post-completion pump observation reporting strictly more delivered insulin before that evidence is consumed. This upgrade shall never occur while the pod reports active delivery or unknown status, before the expected physical completion time (the post-accept early-read shape), for an unproven or different pod identity, or while the register reports a nonzero undelivered remainder; a canceled or interrupted bolus therefore retains its observed partial delivered amount — and a delivered figure shall never be raised on assumption alone. After an issued-dose request step has been consumed into replay or live algorithm input, runtime shall not feed that same request step to the live algorithm again in the same session, regardless of whether a later pump observation reports different requested or delivered units; any same-request-step unit mismatch shall be recorded as discrepancy evidence and telemetry rather than treated as fresh insulin delivery. If fallback-basal replay is also pending and reconnect recovery has a same-pod cumulative pump-delivered insulin delta that includes both the pending issued dose and fallback basal exposure, runtime shall first partition the known pending issued-dose requested amount from the pump total, persist that partition as issued-dose reconciliation evidence, use only the residual pump delta for fallback-basal credibility and replay allocation, and retain the raw pump total separately for telemetry/operator review. If the pump total cannot cover the pending issued dose, the pending dose cannot be matched to the same pod/request, or the residual does not credibly match modeled fallback exposure, runtime shall not guess the split, shall not store or display the raw pump total as actual fallback-basal delivery, and shall suppress replay under ambiguous/unresolved reconciliation. If fallback-basal replay is also pending, issued-dose replay shall be monotonic and non-duplicating with fallback replay. While the pod still reports active delivery or unknown status, or no post-request pump-status evaluation exists yet, runtime shall keep the attribution pending and skip live advancement. Once a fresh settled post-request pump status exists and the original pod has not been proven replaced, matching evidence that is unavailable, mismatched, non-credible (delivered units outside `0...requested`), or missing pod identity shall resolve as assumed delivered per clinical

policy: runtime shall record exactly the requested units as the assumed delivered amount, feed that amount into the first eligible attribution step (by replay or live-step input, under the replay rules above), retain the assumed-vs-confirmed distinction in UI and telemetry, and schedule meal-adaptation exclusion for meal-linked assumed doses so algorithm meal learning does not adapt on assumed evidence. Clinical rationale: under-crediting delivered insulin risks re-dosing insulin already given, so assuming the full requested amount is the conservative direction for insulin-on-board accounting while preserving dosing liveness; the SRS-MEAL-012 user-confirmed unavailable-pod escape is the explicit-operator variant of this same disposition. On relaunch or reconnect with a reachable pod, runtime shall not create assumed-delivered evidence from cached same-pod idle/suspended status, different-pod status, or unresolved-idle status until at least one fresh pump-status refresh has completed for that pod; known old-pod unavailable/no-active/service-stop escape paths retain existing liveness behavior. If a different/new pod is observed before reconciliation, runtime shall record

`different_or_new_pod`, preserve the current algorithm session, assume the issued dose was delivered, feed the assumed delivered amount into the first eligible attribution step by replay or live-step input, clear the old-pod pending attribution after that consumption, scrub unrelated replacement-pod `lastDelivery` from the resumed live algorithm input, and shall not block later insulin-adding automatic commands solely because the original pod cannot be reconciled. For the SRS-MEAL-012 user-confirmed unavailable-pod escape, runtime may consume assumed-delivered clinical-policy evidence without pod identity only when request step, requested units, delivered-unit bounds, and non-active pump status match. Legacy persisted `different_or_new_pod_unresolved` holds shall be treated as nonblocking under this policy. Controlled clarification (v1.52; supersedes the broader fresh-completion exception sentence above): a zero `bolusNotDelivered` register after relaunch shall not by itself override persisted partial-delivery evidence. Runtime shall persist issued-dose observation context when known. A larger settled same-pod observation may replace a partial pump observation only when the persisted context explicitly identifies that partial as capped while delivery was still active. Persisted canceled partials and legacy partials without that positive in-progress provenance shall remain authoritative. Assumed-delivered evidence already records the full requested amount and may be accompanied by a matching fresh pump observation only without increasing the amount attributed to the algorithm. Controlled clarification (v1.65): if the physical bolus command path starts locally but returns a definitive blocked/not-issued outcome, runtime shall clear matching pending attribution and instruct the pump adapter to invalidate only the matching request-step/requested-unit cache. That rejected request shall not later be reconstructed from `bolusNotDelivered` as delivered insulin, displayed as delivered insulin, or fed into any algorithm step. Uncertain outcomes shall retain the cache and attribution for reconciliation.

- SRS-PUMP-011: Every pump command and status await shall be bounded by an app-level watchdog (120 seconds, well above the transport's bounded worst case) that fails the command as uncertain (or the status read as blocked) instead of stalling the serialized loop

work pass; a late transport completion after the watchdog fires shall be ignored safely. The synchronous-completion contract of the pump-event storage delegate callback shall be regression-pinned.

Meal Announce

- **SRS-MEAL-001:** Meal announce shall only use borrow execution when request time is before the next scheduled slot and within borrow window (≤ 2 future 5-minute slots).
- **SRS-MEAL-002:** Meal announce shall be blocked unless an active pod is present, pump status is known, and pump delivery state is confirmed idle. Unavailable-reason precedence shall report `noPump` when no active pod is present, reserve `signalLoss` for active-pod unknown-status conditions, report `pumpDelivering` when an active connected pod is currently delivering even if issued-dose or meal-delivery attribution is also pending, and report `pumpSuspended` when insulin delivery is suspended; only an idle unresolved attribution shall use reconciliation/reconnect guidance. A suspended rejection shall occur before accepted-meal persistence or either algorithm executes, shall issue no pump command, and shall not create a meal recommendation or delivered-dose chart entry. A fresh pump-confirmed transition from suspended to idle shall clear this availability block without requiring another algorithm step. Meal announce shall additionally be blocked while the primary algorithm is in forced-open-loop mode (`openLoopMode == 2`, backup basal running during a released CGM outage), because every pump command is blocked in that state and a meal bolus cannot deliver. The app shall also block before dispatch when the exact prospective meal step would exceed the characterized 12-step glucose-blind tolerance and therefore enter forced-open mode, even if the prior persisted algorithm output was still closed-loop. The app shall persist the algorithm step that consumed each qualifying CGM or fingerstick, use it for this prospective check, and repeat the check in the runtime coordinator after exact step selection but before persisting accepted meal state or executing either algorithm. A current fresh CGM input or a valid persisted $20 \dots 600$ mg/dL manual BG targeted to that exact prospective step permits the combined step; a stale, future-step, invalid, or missing BG shall not bypass the block, and no glucose state shall bypass pump, pod, signal-loss, fallback-reconciliation, delivery, suspension, or request-in-progress gates. A prospective or established forced-open rejection shall map to `backupBasalRunning`, shall not create pending meal state or a delivery-progress state, and shall tell the user that the meal was not announced and no meal insulin was delivered. Availability shall clear automatically once a qualifying fingerstick or sensor reading permits closed-loop stepping.
- **SRS-MEAL-003:** Meal announce UI shall provide explicit unavailable reason and next retry time when applicable, including reconnect/recovery-required guidance when masked fallback may have become active.
- **SRS-MEAL-004:** If meal announce is requested after the next scheduled slot is already due/missed, runtime shall execute meal input on the current due step (`expectedStep`) instead of rejecting solely due to late borrow timing.

- **SRS-MEAL-005:** Meal announce shall be blocked until the loop establishes the first successful anchored step (`firstSuccessfulStepAt`), even when pump and CGM are otherwise available.
- **SRS-MEAL-006:** If the meal announce composer remains visible while runtime availability changes, the app shall revalidate meal availability on runtime events, foreground refresh, and immediately before submit; qualifying-glucose publication and live pump-status changes shall be revalidation triggers so stale inline guidance cannot remain after the condition changes. Unavailable state shall block dispatch and present current blocked messaging instead of using stale availability. During revalidation of an open composer, a stale observed step whose fresh availability check is available shall re-arm the composer in place (selections preserved, observed step updated, transient notice shown) rather than dismissing it; genuinely unavailable conditions shall render current blocked messaging inline within the composer with selections preserved, and shall clear automatically when a later revalidation reports availability. If the open composer was unlocked by a pending fingerstick BG during a released CGM outage (the SRS-MEAL-002 exact-step carve-out) and a loop step consumes that BG before submit, the app shall treat forced-open (backup-basal) availability as transient continuity while the pending BG is cleared but the exact captured step's qualifying-glucose marker has not yet published, even if an intermediate pre-command checkpoint already exposes that step as executed. The composer shall remain armed with selections preserved, shall present at most a passive waiting notice stating the fingerstick BG is being applied and the meal will go with the next dosing step, and shall not instruct the user to enter another fingerstick BG within the accepted BG's validity window (an accepted fingerstick restarts the full CGM-dropout tolerance window). Publication of that qualifying-glucose marker shall revalidate and resume the standard recovery flow without dismissing the composer; a live connected `.delivering` pump status shall instead promptly replace stale backup-basal guidance with pump-busy guidance. The same waiting hold shall apply to a backup-basal-blocked submit issued during the in-flight window. If forced-open availability persists after the qualifying marker has published, or a later executed step proves the captured target passed without qualification, the generic backup-basal messaging shall apply. Submit-time blocking behavior is otherwise unchanged: a blocked submit shall never dispatch.
- **SRS-MEAL-007:** Meal announce submit shall not be presented or logged as success until runtime/coordinator result is known. Executed success shall only be shown after runtime completion, and blocked/rejected outcomes shall produce explicit user-visible result messaging. Uncertain command outcomes shall produce explicit blocked guidance rather than optimistic success.
- **SRS-MEAL-008:** Once a meal announce request is accepted for execution, the app shall persist pending meal-request state across relaunch until the request is explicitly cleared, reconciled against executed step history, or reset by session reset/disarm. Persisted state shall include the concrete target step accepted by runtime and the correlated meal flow identifier needed for lifecycle telemetry closure.

- **SRS-MEAL-009:** While a persisted prior meal request remains pending, additional meal announces shall be blocked after relaunch and the user shall be informed that a meal request is already in progress. If the prior request has an uncertain command outcome, duplicate meal entry shall remain blocked until reconciliation or session reset/disarm, and the user shall be informed that the prior meal delivery outcome is uncertain.
- **SRS-MEAL-010:** If a competing trigger or slot conflict prevents meal execution on the originally observed slot, runtime shall not silently lose or reinterpret the meal request. The app shall block the submit with explicit slot-conflict/retry guidance rather than silently reassigning the meal to a different borrowed step.
- **SRS-MEAL-011:** If meal announce is requested while pump bolus delivery is already in progress, the app shall present a destructive cancel-delivery path on Home that can cancel the current bolus, report the actual insulin delivered before cancellation in Home's standard alert-summary region, and preserve that delivered amount for subsequent algorithm-step accounting when the operator later reopens meal announce.
- **SRS-MEAL-012:** If a meal-announcement delivery-progress modal remains unresolved because the original pod cannot be confirmed, the app shall provide an explicit user-confirmed pod-replacement escape only for the matching meal-linked issued-dose attribution. The escape shall become available when the original pod is known unavailable, past hard service-stop, known different/replaced, or when the expected physical completion time has elapsed by at least 2 minutes without matching evidence. On confirmation, runtime shall record assumed-delivered issued-dose evidence using the original requested units and pod identity when available, record an auditable `user_abandoned_unavailable_pod` disposition, clear only the meal-progress UI fields, preserve the issued-dose attribution for replay/live-step insulin accounting, and route the operator to pod setup. The escape shall not resolve correction-only or basal-only pending issued-dose attributions.

Team review note:

- **SRS-MEAL-004** is accepted for the current software baseline as the implemented late-submit meal behavior.
- Meal `tooLate` retry messaging shall point to the immediate next due-step time boundary (not an added fixed 5-minute delay).
- **SRS-MEAL-007** through **SRS-MEAL-012** are now implemented in the app baseline through runtime-result-gated meal success handling, blocked/rejected submit messaging, persisted pending meal-request state + flow correlation, uncertain-outcome duplicate blocking, relaunch reconciliation against executed-step/pump-delivery evidence, explicit slot-conflict blocking when another step advances after the meal flow was opened, a destructive cancel-delivery flow when meal entry is attempted during active bolus delivery, user-confirmed pod replacement for unconfirmed old-pod meal delivery, and correlated `accepted/resolved` lifecycle telemetry closure.

CGM-Outage Sustained Dosing (BG-Run)

Clinically approved 2026-07-13/14. The controlled behavior is specified by SRS-OUTAGE-001..006, allocated in the Software Design Description, assessed under RA-018, and verified by the mapped SVVP/RTM controls.

- SRS-OUTAGE-001: During a CGM outage the runtime shall maintain a persisted BG-due schedule anchored to the last qualifying glucose (an accepted CGM reading or an accepted fingerstick BG): next BG due = qualifying-glucose time + 120 minutes, or + 60 minutes when the measured value is >200 or <80 mg/dL. Each newer qualifying glucose re-anchors the schedule; the context shall persist across relaunch.
- SRS-OUTAGE-002: While a CGM outage is active the Home screen shall show the next fingerstick-due time (countdown line), including whether backup basal is running; the line clears on CGM recovery. A CGM outage is active for these surfaces when an executed step ran without an accepted CGM value AND any of: the algorithm reports its CGM-dropout offline mode (`openLoopMode == 2`), the step was fed an accepted fingerstick BG, or the persisted schedule's last qualifying glucose was a fingerstick (BG-run in progress). An isolated blind step with the dropout clock intact (e.g., a relaunch or wake step firing ahead of a healthy sensor's next reading) is not an outage and shall not show the countdown.
- SRS-OUTAGE-003: Outage alerting shall be tiered: an actionable "BG due soon" tier pre-scheduled as an OS notification at 15 minutes before the due time; an informational "backup basal running" note while the bridge is released; and a safety-critical "BG overdue" tier pre-scheduled at the due time. The due-soon and overdue tiers shall visibly emphasize the separate Home BG-entry control, shall fire even when no loop step executes near the deadline, and shall be replaced on re-anchor and cleared on CGM recovery or session end.
- SRS-OUTAGE-004: Backup-basal bridge: on a connected forced-open executed step with the 0 U/hr mask armed, the runtime shall actively cancel the mask (the pod's programmed fallback basal resumes immediately) and record the released-for-outage state; mask renewals shall be suppressed while released; on the next non-forced-open executed step the mask shall be re-established BEFORE that step's pump command. A failed cancel keeps the mask (existing forced-open alerting covers the participant) and retries on subsequent forced-open steps; a failed re-arm blocks that step's insulin-adding command (never dose on top of running fallback basal) and retries.
- SRS-OUTAGE-005: There shall be no automatic maximum-outage stop: dosing is sustained indefinitely by qualifying BGs (sponsor/clinical decision 2026-07-13/14; deliberate divergence from the iLet 48/72-hour cutoffs, rationale recorded in RA-018).
- SRS-OUTAGE-006: Fallback basal delivered while the bridge is released shall be accounted for in the algorithm's insulin accounting using the mechanism characterized against the real Algo2015; no new reconciliation evidence class is introduced. Connected released steps shall be credited live, one synthetic per-step pump tuple per executed step: the tuple echoes the algorithm's actual previous ask (retiring the blocked mode-2 ask) and reports the identity-guarded pod-counter delta as delivered at its actual tick timing (no schedule smoothing); per-

step credits accumulate in runtime state for the window. A pod-away gap within a released window (one or more due steps not executed - skipped for unavailable pump status or never attempted while the phone was dark - followed by a pump status carrying pod totals whose pod identity matches the preserved baseline) shall NOT be credited as a single reconnect-step tuple: it shall be reconciled by the standard bounded fallback replay (sponsor direction 2026-07-18, field incident) - the authoritative total is the identity-guarded pod-counter delta against the preserved baseline, distributed across the missed steps at per-step weights from the persisted programmed fallback schedule over each missed step's real slot (weights evaluated in the engaged event's recorded profile timezone, device zone fallback, matching the classic replay path), with per-row shares allocated as whole multiples of the algorithm's 0.01 U insulin-ledger resolution by schedule-weighted largest-remainder allocation - zero-share rows permitted - so the algorithm's booked row total equals the measured delta exactly (the C++ books each row's delivered at 0.01 U resolution; fractional shares would inflate its ledger), replayed as CGM=-1 rows under the SRS-RUN-006 echo-ask convention with the first row echoing the persisted pre-gap ask; a zero-delta gap shall still replay (zero-delivery echo rows retract the window's offline asks); replay rows shall not issue pump commands, and the released mode shall persist through the replay. The synthesized plan shall carry its credit context - the measured delta, the re-anchor counter and pod identity, and the first-row echo ask - in persistent state, and consuming the plan shall apply that context regardless of which work pass executes the replay, so an app termination at any intermediate persistence point between synthesis and consumption neither loses nor double-counts the delta. After the replay, the pod baseline shall re-anchor to the reconnect counter, the measured delta shall accumulate exactly once, and the resumed live step's credit tuple shall therefore compute a zero delta (no double count). An identity mismatch at reconnect shall not replay: the live tuple retracts the persisted ask with zero credit and the baseline re-anchors to the observed pod. At mask re-arm, the release window shall be quantified on the engaged fallback event record - window duration plus the modeled exposure integrated over the pod's programmed fallback schedule - upgraded to a pump-measured confirmed total when per-step credits were accumulated, and labeled `resumed_without_reconciliation` (modeled only) when no credit observation landed.

State Persistence and Recovery

- **SRS-STATE-001:** The app shall persist algorithm state and runtime cadence state across relaunch.
- **SRS-STATE-002:** `Same-Participant Reset` (formerly `Reset Algo`) shall clear persisted algorithm state, persisted cadence state, and step timeline/session history to start a fresh session.
- **SRS-STATE-003:** Pump and CGM manager state shall persist across relaunch to avoid forced re-pairing.

- **SRS-STATE-004:** When masked fallback is armed, runtime persistence shall retain the original programmed basal schedule, the currently programmed fallback schedule/rate, safety target, last known pod identity, last known pod total-delivery baseline/timestamp, last mask-renewal / schedule-refresh timestamps and outcomes, any restore-pending, restore-failed, or reconciliation-required recovery context needed for safe connected maintenance and later disarm/restore, and any pending pump-delta reconciliation state needed for reconnect recovery after offline mask expiry, confirmed/corrected missed-step algorithm replay, or explicit ambiguous unreconciled resume. Runtime persistence shall also retain separate primary and secondary/safety q5 nominal-basal profiles, with local-time slot observations, so fallback schedule generation is traceable to the safety-track nominal basal values rather than the primary dosing track. Runtime persistence shall retain pending issued-dose attribution, the most recent consumed issued-dose delivery, the most recent issued-dose delivery discrepancy, and the most recent issued-dose reconciliation failure disposition needed to avoid guessing or double-feeding delivered insulin across relaunch or pod replacement; stale consumed/discrepancy/failure records shall have a defined expiry path. Meal-linked pending issued-dose attribution shall retain the algorithm-visible request step when available so assumed-delivered meal-learning exclusion can be matched to the algorithm's time base rather than a drifted runtime label.
- **SRS-STATE-005:** Algorithm start, stop, reset, and replacement algorithm-session creation shall occur only in response to an explicit operator action. Runtime recovery, pump reconnect, pod replacement, CGM recovery, fallback reconciliation failure, app launch, app foreground, scheduler wake, telemetry replay, or cloud/auth state shall not independently stop the algorithm, start the algorithm, reset the active algorithm session, or create a replacement session. Recovery logic may block command application, suppress replay, mark missed steps as unreconciled, and continue the existing session at the current cadence slot.

Algorithm Verification Controls

- **SRS-ALG-001:** The Algo2015 implementation and bridge path shall maintain a deterministic golden-vector regression suite with fixed inputs and expected outputs under version control.
- **SRS-ALG-002:** Algo2015 verification shall include structural coverage reporting for C bridge and C++ algorithm sources, with predefined coverage targets and documented rationale for uncovered regions.
- **SRS-ALG-003:** Bridge contract behavior for null inputs, state reset edge conditions, sentinel mappings, and subject-id boundary handling shall be explicitly verified.
- **SRS-ALG-004:** Algorithm state continuity shall be verified across multi-step execution, persistence/reload, and reset-to-fresh session boundaries.
- **SRS-ALG-005:** Any change to pregnancy configuration inputs (`target`, `meal upfront`, `TMAX`) shall include differential algorithm-output evidence against baseline replay vectors.
- **SRS-ALG-006:** Formal Algo2015 verification runs shall include a static-analysis lane for algorithm/bridge sources with logged findings status and run linkage metadata.

- SRS-ALG-007: MISRA assessment for the host-side investigational Algo2015 lane shall be risk-based and conditional: each formal run shall include either linked MISRA evidence with signed deviation handling (if applicable) or an explicit not-applicable decision rationale with compensating controls.
- SRS-ALG-008: The runtime shall host the primary dosing track and the secondary/safety track on isolated algorithm instances such that neither track's execution can read or advance the other's algorithm state (buffers, counters, adaptation history, or persisted state). Instance isolation shall be regression-verified, including the primary track's CGM-dropout tolerance measured in its own executed steps.
- SRS-ALG-009: For every insulin-adding algorithm step, the host pump recommendation shall equal the sum of Algo2015 `bolusInsulinRequested`, `basalInsulinRequested`, and `mealInsulinRequested`. The persisted issued-dose request and later pump-feedback `unitsRequested` shall use that same total so a completed delivery can satisfy the Algo2015 delivered-insulin reconciliation gate without omitting or double-counting any component.

Telemetry and Traceability

- SRS-LOG-001: For each executed step, telemetry shall include an explicit step execution timestamp (`step_executed_at`), algorithm input snapshot, output snapshot, and pump command result. The app shall retain a complete protected, backup-excluded step archive for the active algorithm session and use that archive for the clinical-gated CSV export; the Recent Dose Steps presentation cache may remain bounded to the newest 576 rows. Explicit algorithm-session reset shall clear the active-session archive. Local Recent Dose Steps review shall identify an executed step whose algorithm input included a valid manual/fingerstick BG value, show the exact fingerstick value used, and retain any concurrent CGM value rather than replacing it. Algorithm output telemetry shall retain Algo2015's read-only delivered-insulin echo fields (`bolusInsulinDelivered`, `basalInsulinDelivered`, `mealInsulinDelivered`, and `deliveryTime`) and cloud loop-step telemetry shall flag delivery-echo divergence when matching reconciled pump input differs beyond pump mechanical resolution. When a pump observation disagrees with already-consumed issued-dose delivery for the same request step, telemetry shall record the consumed-vs-observed discrepancy fields while suppressing the observation from algorithm insulin input. If an assumed-delivered meal-learning exclusion expires without algorithm acknowledgement after the analysis window, step telemetry shall include dashboard-visible disposal evidence and shall not require a participant-facing alert.
- SRS-LOG-002: Telemetry storage shall support export with stable column definitions.
- SRS-LOG-003: Telemetry persistence shall not perform CSV or recovery-ZIP generation, file copying, or compression synchronously on the main actor; export work shall use staged asynchronous processing so the participant UI remains responsive.
- SRS-LOG-004: The default cloud-log upload threshold shall be `Error` with inclusive evaluation (`selected_level` and higher severities). Upload level is raised only by a server-issued remote logging config (SRS-LOG-011); no in-app logging control is exposed in any build (the former

debug-build threshold and integration-session controls were removed 2026-07-16, and the build-designated profile was superseded the same day - study logging is configured server-side only).

- SRS-LOG-005: Clinical settings save flow shall emit `ui.critical` telemetry for `state_viewed`, `submit`, `cancel`, and `blocked` outcomes with stable `screen_id/element_id` values and old/new field details for changed clinical parameters.
- SRS-LOG-006: `app.lifecycle.launch`ed and `app.lifecycle.foregrounded` telemetry payloads shall include timezone and clock-check context fields: `device_timezone_id`, `device_utc_offset_seconds`, `clock_check_result`, and when available `clock_check_skew_seconds`, `clock_check_rtt_ms`, `clock_check_at_utc`.
- SRS-LOG-007: Meal-announce telemetry shall expose a correlated request lifecycle so cloud reconstruction does not rely on optimistic submit success alone. Implemented baseline includes `submitted`, `accepted`, `success`, `blocked`, `uncertain`, and `resolved` transitions keyed by meal `flow_id`, with replay state persisted until emission so `accepted` and `resolved` are not lost across terminate/relaunch windows; `resolved` is emitted for immediate completion, relaunch/pump-evidence reconciliation after uncertainty, or session-reset clear. Loop command telemetry shall preserve command outcome semantics, including uncertain delivery, rather than collapsing all non-success outcomes into generic blocked state.
- SRS-LOG-008: Target-selection telemetry shall capture clinician-controlled regular-settings target profile changes and participant target-change approval-capture events with stable `ui.critical` event identifiers and required detail fields (`target_range_profile`, requested/applied target, approval metadata, and blocked/cancelled reasons where applicable).
- SRS-LOG-009: Masked fallback review telemetry shall capture fallback arm, maintenance deferred, mask renewal, schedule-refresh unchanged checks, schedule refresh, offline mask expiry, and restore success/failure with selected fallback rate, source, safety target, duration, reconciliation status, pod-continuity result, and failure detail sufficient for Home Recent Dose Steps review. Resolved reconnect-recovery events shall also retain modeled fallback delivery and, when available from proven same-pod continuity, pump-reported delivered insulin since the last connected fallback baseline. Per-step local review telemetry shall also retain the executed primary and secondary algorithm step summaries needed to display each track's step count, suggested dose, nominal basal, and instant basal in Recent Dose Steps. When confirmed/corrected same-session fallback delivery reconciliation occurs, local review/export shall include only fallback-active missed-step primary/secondary algorithm replay rows allocated from the credible same-pod pump-reported delivered-insulin delta using the persisted programmed fallback schedule as the weighting model; each replay step shall use unavailable CGM input (-1), that step's delivered-insulin allocation including explicit 0 U input when no pump allocation belongs to that fallback-active step, and no pump command application. Pre-activation disconnected missed steps shall remain absent from replay telemetry rather than appearing as synthetic 0 U rows. The resumed live step shall use actual refreshed pump status and shall not receive a duplicate aggregate recovered-delivery input. Cloud telemetry shall additionally emit a dedicated `loop.fallback.event` family with a

stable `fallback_event_id`, lifecycle kind, structured delivery/reconciliation summary fields including `pod_continuity`, and explicit occurrence timestamps so BionicScout can render fallback-related events separately from ordinary loop-step telemetry and distinguish modeled fallback delivery, pump-reported delivered insulin, and reconciled delivery when available. Fallback arm and schedule-refresh cloud events shall include the programmed four-entry schedule, safety q5 profile source/kind metadata, observed/imputed slot counts, per-six-hour-segment observed-slot counts (four counts, each out of that segment's 72 five-minute slots), profile timezone metadata, and the 288 q5 profile rates used for that specific schedule-programming operation. The steady-cadence `schedule_checked_unchanged` maintenance event shall carry the same programmed four-entry schedule and profile coverage metadata (source/kind, observed/imputed slot counts, per-segment observed-slot counts, timezone metadata, and the profile last-updated timestamp) while omitting only the bulk 288-rate array, so the active fallback schedule and its provenance remain visible in recent telemetry and in Recent Dose Steps even when a stable profile produces no arm/refresh event for days (sponsor request 2026-07-20); routine mask renewals shall continue to carry no schedule/profile payload. Recent Dose Steps fallback schedule rows (arm, schedule refresh, and unchanged check) shall show the four programmed segment rates, a plain outcome label distinguishing armed/updated/unchanged, and per-segment provenance that marks a segment as imputed when fewer than half of its 72 slots are observed.

- **SRS-LOG-010: Superseded 2026-07-16 (sponsor direction), never shipped to a study participant.** The build-designated cloud-log profile (Info.plist-baked level/end-date/label establishing an on-phone integration logging session) is removed: the server-issued logging config (SRS-LOG-011) is the sole study logging lever, and the app reads no session or profile state (storage left by pre-removal builds is ignored). Requirement text retained in revision 1.34 history for traceability; ID retired, never repurposed.
- **SRS-LOG-011:** The app shall fetch the server-issued logging config (`GET /v1/config/logging`, authenticated) at launch, on each foreground, and periodically (nominal 30-minute interval) while running, all outside UI-test mode, and apply it as a remote upload-level override carrying the server-supplied expiry. The server-issued config is the **sole** study logging lever: no build-baked profile or on-phone logging session exists, and the upload threshold is the active override's level, else the default (SRS-LOG-004). A `null` or expired server config shall clear any stored override; a malformed response shall leave the stored override untouched; fetch or auth failures shall never block or alter app behavior (the last-applied override keeps applying until its own expiry). When a fetch changes the stored override, the app shall log a threshold-exempt cloud marker (`remote_logging_config_applied` with the config level/expiry/reason, or `remote_logging_config_cleared`); unchanged re-fetches log nothing. The config is issued only from the BionicScout admin surface (dashboard scope + clinician/admin group). Settings shall present the effective logging state (level, expiry, and whether it comes from the study team or the default) as read-only status text with no controls.

- SRS-LOG-012:** Every uploaded telemetry envelope shall carry a durable monotonic `sequence_no` assigned by the persistence ledger at enqueue (the uploaded bytes and durable row shall be structurally unable to disagree), plus `install_id` and `session_id`. Events posted outside the ledger (e.g. dropped-events markers) shall reserve a durable sequence number so a failed send is server-detectable as a gap. Sequence numbers enable BionicScout to prove per-install delivery completeness; events from earlier builds without `sequence_no` are evaluated on a steps-only basis and shall not register as gaps. The protected, backup-excluded SQLite outbox shall transactionally and repeatably migrate any existing file-spool backlog during an update, including a later fallback spool created after a transient ledger-open failure, and delete source data only after committed import. Retained clinical/runtime entries shall not be evicted by an arbitrary count cap. Telemetry emission shall return after durable local commit and schedule network drain independently; a newer immediate drain shall not cancel an older event's scheduled retry. Recovery shall deliver retained evidence before bounded debug/log chatter while preserving sequence numbers and per-class FIFO order. Chatter evicted at its class cap shall be counted without immediate per-event network work and summarized in one aggregate dropped-events marker after transport progress resumes. The client may upload up to 50 unchanged persisted envelopes and 512 KiB through an additive batch route with per-event acknowledgement; unsupported batch routes shall return every selected row to pending and fall back to the existing single-event route without loss or duplicate clinical effect. New Participant Reset shall physically recreate the outbox database and remove its WAL/shared-memory sidecars before participant-scoped reset completion, while preserving only install-scoped monotonic sequence metadata. Repeated notification clear requests shall emit lifecycle telemetry only for a persisted scheduled-to-cleared transition. Settings shall provide passive, non-alerting visibility of pending and permanently rejected study-data records without changing dosing or reset policy. A permanent client-side telemetry rejection (HTTP 4xx other than rate limiting) shall remain in the protected local outbox as failed-permanent evidence carrying the event type, sequence number, failure timestamp, HTTP status, and error detail, and shall remain visible to BionicScout as a sequence gap. The app shall not surface a generic participant-facing alert for that non-actionable transport condition; a subject-identity conflict shall retain its specific actionable alert without also producing a generic upload alert, and HTTP 429 shall remain retryable. A one-time launch migration shall clear any orphaned generic telemetry-rejection notification from an earlier build, and loading persisted alerts shall always remove an earlier-build generic telemetry-rejection alert and its scheduled notification.
- SRS-LOG-013:** Temporary Target telemetry shall emit durable lifecycle events for activation, first algorithm use, replacement, early end, expiry, session stop, profile change, and same-participant/session reset with a stable override identifier, subject, temporary/permanent/safety targets, duration, absolute start/end timestamps, and first-applied step when applicable. New Participant Reset is the intentional exception: it shall erase the outgoing participant identity, Temporary Target state, and pending lifecycle records without emitting a new outgoing-participant lifecycle event. Every loop-step telemetry payload shall expose the permanent target, target source, temporary-target identifier/expiry, whether the override was actually

applied to that step, and the safety target. BionicScout shall project the lifecycle and applied-step evidence into an operator-visible state that distinguishes an active target awaiting first use from one observed in algorithm use, preserves terminal state against delayed/out-of-order events, and does not change telemetry wire-field types.

- SRS-LOG-014: G7 replacement acquisition shall emit stable, deduplicated telemetry for ten-minute acquisition stall and successful sensor adoption. Evidence shall include the acquisition event, trigger, previous sensor identity when available, new sensor identity on adoption, and event time. Successful durable enqueue shall be marked in persisted acquisition state so repeated manager sync or relaunch cannot reserve an additional wire sequence for the same event; failed enqueue shall remain retryable. Until BionicScout accepts dedicated sensor-acquisition event types, the app shall carry these payloads through the accepted `cgm.state.changed` family without changing wire-field types.
- SRS-LOG-015: The clinical-unlock-gated share action in Recent Dose Steps shall produce one recovery ZIP containing the complete retained step-telemetry CSV and byte-faithful copies of the applicable legacy Algo2015 run artifacts, without modifying their writer or lifecycle. Timestamped `BP_<epoch>.txt` and `BP_LOG_<epoch>.txt` files shall be grouped by session epoch; the fixed-name `BP_ReadMatrix.m` shall join only the epoch it references. When the active algorithm inspection epoch is known, only that exact group may join the CSV; if the exact group is unavailable, export shall remain CSV-only rather than attach another run. When no inspection epoch is available after relaunch, the newest valid grouped epoch may join the CSV. Export shall copy every included source only through a protected, backup-excluded temporary snapshot before creating the ZIP and shall never archive a live path directly. The CSV is required; any subset of the three algorithm files may accompany it. An unavailable CSV shall expose no share item and shall never produce an empty archive. The archive name shall include a sanitized subject-or-device identifier and export timestamp. Collection, copy, and archive generation shall execute off the main actor, sharing shall use the existing system `ShareLink`, and temporary export data shall be removed when the Recent Dose Steps view closes.

UI Safety and Status

- SRS-UI-001: Home loop-status card shall use deterministic state precedence: `No Session > No Pod > No CGM > Ready > age-based states`, where age-based states are derived from cadence due-time (`firstSuccessfulStepAt + (lastExecutedStep + 1) * 300s`) rather than raw `lastSuccessfulRunAt`.
- SRS-UI-002: Home availability and status messaging shall align with runtime behavior (no false available state for blocked actions). While the pump reports suspended delivery, Home shall display `Insulin delivery is paused`. independently of the reminder interval. The Home Manual BG and Meal Announcement primary actions shall remain visible and actionable in a bottom safe-area control surface while Home status, alert, and chart content scrolls; persistent placement shall not bypass or change either action's existing availability policy.

- SRS-UI-003: Initial CGM and Pod startup/setup modals shall provide an explicit `Cancel` action that dismisses the modal without forcing entry into settings.
- SRS-UI-004: If the meal announcement composer is visible and the app transitions to background, the composer shall auto-dismiss/cancel so meal intent does not persist across lifecycle interruption.
- SRS-UI-005: If the latest CGM reading is older than 11 minutes, or if the latest CGM reading is present but not reliable (`hasReliableGlucose == false`), CGM display surfaces shall mask the reading as `--` and shall not display a trend arrow.
- SRS-UI-006: The app shall perform UTC drift checks on launch, on timezone/significant-time-change events, and on foreground when the last successful check is older than 24 hours. If absolute clock skew exceeds 600 seconds, the app shall present a non-blocking actionable warning no more than once per 24 hours; unavailable UTC checks shall not show warning spam.
- SRS-UI-007: CGM display surfaces shall render boundary readings as textual values `LOW` (`<=39 mg/dL`) and `HIGH` (`>=401 mg/dL`) instead of numeric mg/dL values.
- SRS-UI-008: Home CGM chart y-axis scaling shall use bounded stepped maxima (300, 350, 400 mg/dL) based on displayed data peak. Glucose samples shall render as discrete, value-colored points at their recorded timestamps without a connector line or interpolated area fill; chart/scrub rendering shall keep boundary-value interpretation consistent with SRS-UI-007.
- SRS-UI-009: The Home investigational badge text shall be server-configurable (`GET /v1/config/banner`, fetched on the shared SRS-LOG-011 cadence: launch, foreground, 30-minute periodic): an absent config shows the built-in default `NOT FOR HUMAN USE` (fail-safe - a fetch failure or unset config can never remove the investigational label), a configured string shows verbatim (`<= 60` characters, issued only from the BionicScout admin surface), and an explicitly blank configured string renders no badge at all (no empty capsule). The BionicScout dashboard chrome mirrors the same three-state behavior.

Clinical Settings and Pregnancy Configuration

- SRS-CLIN-001: The app shall provide a dedicated `Clinical Settings` area for clinician-only configuration and control actions.
- SRS-CLIN-002: After a usable subject clinical configuration has been saved, entry to `Clinical Settings` and every subsequent clinical-setting edit shall be gated by a subject-scoped offline clinical unlock verifier. The only exception is initial configuration while no usable subject configuration exists, as defined by SRS-CLIN-013; that exception shall close immediately after the first successful save. The app shall provide a secure local material-installation path for provisioned verifier material; installation shall validate the subject/material match, persist verifier material in local secure storage, preserve the last locally accepted counter across material refresh, clear any active unlock expiration on material refresh, and reset local failed-attempt/lockout state. After verifier material is provisioned into local secure storage, code verification shall not require internet access. The verifier shall normalize grouped ASCII numeric code input, search only counters greater than the last locally accepted counter

through the provisioned offline lookahead limit, compute the HMAC code using the provisioned current/supported secret version, persist the matched counter before opening clinician controls, set a local unlock expiration timestamp for the provisioned unlock duration, and reject malformed codes, non-ASCII digits, same/lower counters, wrong-subject codes, counters beyond lookahead, and unsupported secret versions. Manual lock shall persist unlock expiration clearing; if that persistence fails, the app shall keep clinician controls visibly unlocked and show a storage-failure message rather than falsely presenting a locked state. The app shall provide an operator-facing provisioning entry (visible while clinician controls are locked) that accepts the subject ID and the BionicScout-issued one-time provisioning code (30-byte payload, Crockford base32 with format version, subject binding check, and checksum), decodes it offline, and installs the resulting verifier material through the same validated installation path; decoding shall reject malformed/truncated codes, checksum failures, unsupported format versions, and subject mismatches. Installed verifier material shall contain only the per-subject derived device secret; the issuing master secret shall never be provisioned to or stored on the device. The app shall additionally support fetching the provisioning code over the authenticated BionicScout session (subject-device token; the backend releases material only for the subject claimed by that account) and shall validate and install the fetched code through the same decode/installation path as typed input, including rejecting responses whose subject does not match the requested subject. Production master secrets shall not be hardcoded in app source.

- SRS-CLIN-003: Subject ID, Weight, Start Algo, and Same-Participant Reset (formerly Reset Algo) controls shall be hosted in Clinical Settings and not exposed in non-clinician settings sections.
- SRS-CLIN-004: Target selection shall support 90, 100, 110, 120, and 130 mg/dL only.
- SRS-CLIN-005: Meal upfront percentage selection shall support exactly 75% and 90%.
- SRS-CLIN-006: TMAX selection shall support 40 . . . 70 minutes inclusive, in increments of 5.
- SRS-CLIN-007: Clinical settings values shall be persisted with versioned defaults/migration so runtime behavior is deterministic across relaunch and app update.
- SRS-CLIN-008: Relocating Start Algo and Reset Algo (since relabeled Same-Participant Reset) into Clinical Settings shall not change their runtime behavior semantics.
- SRS-CLIN-009: Clinical Settings shall provide a clinician-controlled target-access profile for participant-facing settings with exactly two values: Pregnancy and Standard.
- SRS-CLIN-010: Participant-facing settings and the clinician target selector shall expose only the target options enabled by the current target-access profile: Pregnancy -> 90, 100, 110 mg/dL; Standard -> 110, 120, 130 mg/dL.
- SRS-CLIN-011: Participant target changes in regular settings shall require approval capture before apply. The app shall block the change unless the operator records the approving clinical staff member name and an approximate approval time.
- SRS-CLIN-012: If the clinician changes the target-access profile in Clinical Settings (outside first-launch setup) while the draft target is outside the newly selected profile subset, the app shall normalize the draft target to the nearest allowed value before save rather than preserving

an out-of-profile target. During first-launch setup, a mode switch instead applies that mode's default target per SRS-CLIN-013.

- SRS-CLIN-013: First-launch clinical defaults shall be the Standard target-access profile with a 120 mg/dL regular target, 75% meal upfront, and 40-minute TMAX (clinical direction 2026-07-14; study insulin Fiasp). Defaults shall apply only to new setups and shall never rewrite previously persisted participant values. Loading an empty configuration store shall return the first-launch defaults WITHOUT persisting them, so the first-launch setup gate stays open until an operator saves. While no usable subject configuration exists, study staff shall be able to review and save the initial clinical configuration without a clinical unlock code; the successful first save shall close that exception, after which SRS-CLIN-002 governs all Clinical Settings access and edits. In first-launch setup, switching the mode shall select that mode's default target (Standard -> 120, Pregnancy -> 110 mg/dL), and the review summary shall include the selected mode.
- SRS-CLIN-014: Clinical Settings shall provide a New Participant Reset, distinct from the same-participant algorithm reset, that erases all participant-scoped data on the device - clinical configuration, algorithm and runtime state, Temporary Target state including pending lifecycle records, local step telemetry and its CSV export, the cloud telemetry outbox, alert history, stored pump and CGM manager state, the subject-scoped clinical-unlock material including burned counters, and onboarding completion/claim - and signs the account out, returning the app to first-launch setup. The reset shall invalidate device-state persistence callbacks and matching in-flight device telemetry before erasing stored manager state so an asynchronous completion cannot recreate prior-participant state after the wipe. The reset shall be refused while a pod is active and shall require explicit destructive confirmation; when an algorithm session is armed, the control shall offer to stop the session first and then proceed to the reset confirmation. Before the erase confirmation, the app shall attempt a final telemetry upload flush and, when undelivered records remain, the confirmation shall state their count and that resetting discards them permanently - the reset itself remains allowed (sites must be able to reset a broken phone). On confirmation, new telemetry transport shall be gated, any ordinary prior-participant transport already in flight shall be allowed a bounded interval to quiesce, and the reset shall fail closed without wiping participant state if transport cannot quiesce or outbox erasure cannot be persisted. The live algorithm-session and active-Pod guards shall be rechecked after asynchronous work, and algorithm arming shall be unavailable while reset owns the transaction. A reset that proceeds with undelivered records shall first erase the outbox durably while preserving the install's telemetry sequence continuity, then initiate a nonblocking best-effort `telemetry.outbox.discarded` marker carrying pending and rejected counts; a failed marker remains server-detectable as a reserved sequence gap (SRS-LOG-012). Resets blocked by their initial armed-session or active-Pod guard shall discard nothing.
- SRS-CLIN-015: Weight entry shall support 50 . . . 500 lbs inclusive; a configuration whose weight is outside this clinically confirmed range shall not be usable for arming. The configuration store shall clamp persisted weight to at most 500 lbs and shall never raise a lower entry (an unset 0 remains unset so first-launch setup stays gated; a sub-plausible

positive value remains visible but unusable). As a last rail, the dosing algorithm layer shall clamp the weight it consumes to 20...230 kg, where the low clamp reduces dose scale (safe direction) and the high clamp bounds runaway dose scale.

- SRS-CLIN-016: The participant-facing `Exercise` and `Insulin Delivery` settings area shall appear above the `User` section and provide a `Pregnancy-only Temporary Target` with explicit, non-preselected target choice (120 or 130 mg/dL) and duration choice (30, 60, 90, or 120 minutes). Target and duration controls shall use the app's standard outlined selected-state treatment. If either choice has been made and the participant attempts to leave before successful review and activation, the app shall warn that the selections will be lost and require explicit confirmation before discarding them. Activation shall require an armed algorithm session, an active Pod, and no pending masked-fallback recovery; it shall issue no immediate Pod command and shall become effective only when a valid algorithm step executes. The override shall be subject-bound, persisted with an absolute expiration timestamp, restored across relaunch, shown with its end time on Home and Settings, replaceable deterministically, and endable early. The first algorithm step at or after expiration shall use the permanent target regardless of notification delivery. Only the primary-controller target shall use the override at each row's actual execution time, including historical replay rows; the safety controller, permanent target, q5 fallback profile, and programmed backup-basal schedule shall remain derived from the permanent Pregnancy configuration. Profile change away from Pregnancy and explicit session stop/reset shall clear the active override without automatically starting, stopping, resetting, or re-anchoring an algorithm session; New Participant Reset shall remove all Temporary Target state. The same combined screen shall directly expose the existing Pod `Suspend Insulin Delivery / Resume Insulin Delivery` command while normal Pod Settings shall omit that duplicate control; unrelated participant-reset guidance shall not be shown on this screen. Suspension shall retain the existing 30, 60, 90, or 120 minute reminder choices; the selected duration shall not automatically resume insulin, and delivery shall remain suspended until the participant explicitly resumes it. The action shall be disabled when no active usable Pod exists or while the Pod is changing suspension state. When a suspension reminder ends, the active alert shall offer a direct `Resume Insulin` action that uses the existing Pod resume command, prevents duplicate in-flight commands, remains retryable after failure, and stays active until the pump alert lifecycle confirms recovery. If insulin remains suspended for 15 minutes after that reminder, the alert shall escalate to safety-critical severity and issue a fresh background notification without automatically resuming insulin. Suspend-notification taps shall continue to route to `Settings > Exercise and Insulin Delivery`.
- SRS-CLIN-017: An explicit Clinical Settings draft transition from `Standard` to `Pregnancy` shall initialize meal upfront to 90%, while retaining 75% as a selectable value before save. Opening settings, canceling without save, relaunching, or loading an existing Pregnancy configuration shall not silently rewrite the persisted meal-upfront value.
- SRS-VAL-001: Weight entry shall be pounds on UI, integer-only, stored as kilograms for runtime/algorithm use; in current UX scope this entry is clinician-gated.

Team review notes:

- SRS-CLIN-002 is accepted for the current software baseline as an investigational single-device clinical unlock control. Broader production authorization/role-based access remains outside this IDE baseline unless explicitly pulled into scope.
- The Pregnancy configuration control set is defined by SRS-CLIN-001..017 and its allocations in the SDD, Risk Analysis, SVVP, and RTM.
- Implemented baseline now includes a clinician-controlled participant target-access profile (Pregnancy / Standard), participant target-change approval capture in regular settings, and a clinician target selector that is constrained to the currently selected profile subset.

User Alerts and Escalation

- SRS-ALERT-001: The app shall normalize alerts from pump (OmniBLE), CGM (G7SensorKit), algorithm/runtime, and app safety policies into a single alert domain model.
- SRS-ALERT-002: Each alert shall include source, severity, timestamp, user-facing message, and recommended action metadata.
- SRS-ALERT-003: Alert presentation shall apply severity-based channels and deterministic precedence so critical alerts are never hidden by lower-severity alerts.
- SRS-ALERT-004: The app shall apply a 5-minute display and alert debounce to transient pump signal-loss and no-active-Pod indications. This presentation debounce shall not delay or alter pump-command, connection-safety, or dosing gates.
- SRS-ALERT-005: Alert life-cycle behavior shall define clear/acknowledge rules per alert type (auto-clear vs explicit acknowledge). Internal reconciliation evidence such as expired meal-learning exclusion disposal shall be routed to telemetry/dashboard review rather than participant-facing active alerts unless a separate safety condition requires user action.
- SRS-ALERT-006: Protocol-required alert conditions and response wording shall match the controlled alert inventory and SDD-POL-008, and each claimed condition shall map to an executed verification identifier in the RTM.
- SRS-ALERT-007: For high-priority non-CGM alerts (actionable/safetyCritical), the app shall issue background local notifications with de-duplication and per-severity cooldown to avoid alert spam.
- SRS-ALERT-008: The app shall provide an in-app Alert Center showing both active alerts and recently cleared alerts.
- SRS-ALERT-009: Pump and CGM alert issue/retract lifecycle state shall persist across app relaunch, including lookup of unretracted alerts and restoration of active alert visibility.
- SRS-ALERT-010: Alerts with time-sensitive countdown wording shall refresh displayed remaining time at least once per minute while active so user-facing timing remains current.
- SRS-ALERT-011: Home shall present active alerts as a severity-prioritized stack (severity first, then condition-specific precedence, then recency) that always shows the highest-priority alert, indicates multiplicity explicitly, and provides an expansion affordance revealing all remaining Home-visible alerts in place. Within the same severity, active ALERT-PUMP-SIGNAL-LOSS shall

appear above `ALERT-ALGORITHM-CHECK-BG-REQUESTED` so a disconnected Pod is not masked by a newer fingerstick prompt; any safety-critical alert shall still outrank both. Generic alerts whose root cause is already named by a specific active alert (per the sponsor-approved suppression matrix in `SDD-POL-008`) may be hidden from the Home surface only; safety-critical alerts shall never be suppressed, and Alert Center shall always show the complete active list. When suppression hides one or more active alerts, the Home stack shall indicate how many related alerts are hidden and shall provide a direct affordance to open Alert Center, so the Home surface and the alert-count badge never silently disagree.

- `SRS-ALERT-012`: Safety-critical pump alerts (`ALERT-PUMP-FAULT`, `ALERT-PUMP-INCOMPATIBLE`) shall not auto-clear solely due to `hasActivePod == false`; they shall clear only on explicit lifecycle retraction/recovery or explicit acknowledge workflow.
- `SRS-ALERT-013`: When armed-loop execution has no successful algorithm step for more than 15 minutes, the app shall issue an actionable interruption alert distinct from G7 unavailable/failed/expired alerts and clear it once successful step execution resumes or the loop is disarmed.
- `SRS-ALERT-014`: The interruption alert shall use broader `Algorithm Stepping Interrupted` semantics, triggered when the armed loop has not achieved a successful algorithm step for more than 15 minutes, regardless of whether the immediate blocker is missing CGM, missing pod, pump signal loss, suspension, fallback-delivery recovery, or another runtime execution gate. Blocker-specific participant copy shall be derived from the latest unresolved runtime outcome or authoritative current pump condition; a broad CGM lifecycle/status alert alone shall not be treated as proof that CGM caused the missed step. Current suspension shall outrank fallback-recovery detail, fallback recovery shall outrank stale persisted CGM or pump-status detail, and a successful live step shall invalidate the persisted interruption blocker and clear the active interruption alert. After a pump-status-unavailable attempt, a pump-status refresh may replace the stale unavailable guidance only when it proves an active Pod has a newer `idle` status; the same alert shall then state that Pod communication is restored and automated dosing is waiting for the next glucose update. This status-only alert transition shall not trigger algorithm execution or a pump command. Stale, unknown, delivering, suspended, or no-active-Pod status shall not claim recovery, and a later unknown status shall restore unavailable guidance.
- `SRS-ALERT-015`: BionicLoop CGM availability/failure alerts shall be in-app informational status only on Home / Alert Center, shall not require explicit acknowledge, and shall not schedule background local notifications. The FDA-cleared Dexcom application shall be treated as the source of truth for CGM alarming and shall be configured accordingly in study use.
- `SRS-ALERT-016`: When a trustworthy G7 reading is $<55 \text{ mg/dL}$, the app shall issue an in-app-only urgent-low review alert on Home / Alert Center, shall allow clinician acknowledge/review capture, shall not schedule background local notifications, and shall auto-clear the active alert state on a later trustworthy G7 reading $\geq 55 \text{ mg/dL}$. This alert shall be app-derived from CGM data and shall not claim Dexcom-app acknowledgement.

- **SRS-ALERT-017:** Any no-active-pod condition shall raise a safety-critical pump alert after a 5-minute debounce and shall repeat background local notification attempts at least every 30 minutes while the no-active-pod condition remains true, until active pod state is restored or explicit alert-lifecycle recovery clears the alert.
- **SRS-ALERT-018:** When Algo2015 output `checkBG` is active, the app shall surface a deduped actionable fingerstick BG prompt with foreground haptic feedback and visible emphasis on the Home BG entry control. The prompt shall clear on manual BG entry or a later algorithm output where `checkBG` is inactive, shall not re-alert while the same active request persists across successive steps, and shall not be restored as stale active alert state after app relaunch.
- **SRS-ALERT-019:** When Algo2015 primary or safety algorithm output `openLoopMode` equals `LOOP_MODE_FORCED_OPEN`, the runtime shall record the algorithm step and telemetry, preserve pre-command fallback-basal maintenance, suppress that step's automated pump command, and keep the session and cadence intact: forced-open is the algorithm's self-clearing CGM-dropout indicator, not a session-fatal condition (sponsor decision 2026-07-08, consistent with **SRS-RUN-007**). The app shall surface a deduped auto-clearing informational note for the condition; the note shall escalate to safety-critical advisory wording only when the condition persists for at least 3 consecutive executed steps. Escalated wording shall instruct entering a fingerstick BG to resume automated dosing plus sensor-check and study-team contact guidance, and shall never demand an algorithm-session reset or claim insulin delivery that is not occurring. The alert shall remain active across skipped/no-telemetry publications and shall clear only after a later algorithm output reports forced-open inactive. While the backup-basal bridge is released (**SRS-OUTAGE-004**), this note is superseded by the outage alert set (**SRS-OUTAGE-003**) and retracted.
- **SRS-ALERT-020:** When an explicit or device-lifecycle-triggered G7 replacement acquisition remains unresolved for 10 minutes, the app shall surface a deduplicated, actionable, auto-clearing in-app alert directing the participant to finish starting the sensor in the Dexcom app and retry from Settings. The condition shall clear on successful adoption or replacement-acquisition termination, shall survive relaunch through persisted acquisition state, and shall not schedule a BionicLoop background CGM notification because the Dexcom app remains the CGM alarm source under **SRS-ALERT-015**.

Alert evolution notes:

- Current implemented baseline is **ALERT-ALGORITHM-STEPPING-INTERRUPTED**.
- The alert triggers on absent successful step execution rather than CGM-specific receipt loss alone.
- **Algorithm Stepping Interrupted** preserves root-cause visibility so stronger underlying pump alerts remain available and prioritized while CGM state remains visible as separate informational context.

Alert source inventory and normalized mapping baseline:

- Alert Inventory and Mapping

Security and Privacy

- SRS-SEC-001: Telemetry architecture shall support secure cloud upload, with delivery provable per SRS-LOG-012, as a supportive monitoring and corroboration function; local dosing shall not depend on cloud availability. The BionicLoop exported CSV shall be the official insulin-delivery outcome source, and the clinical-gated exact-session recovery ZIP shall be the primary software session-reconstruction record for its covered interval. Local CSV and legacy Algo2015 source and staging artifacts shall use the specified complete or complete-until-first-unlock protection, shall be excluded from backups, shall not be exposed through OS file sharing (`UIFileSharingEnabled` and `LSSupportsOpeningDocumentsInPlace` shall remain absent), and shall be accessible solely through clinical-gated in-app share flows. No software baseline shall permit ungated local file access.
- SRS-SEC-002: The telemetry spool shall use complete-until-first-user-authentication file protection and backup exclusion. App-enumerated Algo2015 diagnostic artifacts shall receive the same protection and backup exclusion without blocking algorithm execution if attribute application fails. CSV and recovery-ZIP staging shall use complete file protection, shall remain unavailable outside the clinical-gated in-app share flow, and shall be removed after sharing or cancellation.
- SRS-SEC-003: Protected cloud API access is deferred from the current software package. If cloud-backed protected APIs are included in a future accepted baseline, they shall require authenticated user identity managed by Cognito before access is granted.
- SRS-SEC-004: Multi-provider onboarding policy is deferred from the current software package. If included in a future accepted baseline, onboarding shall define the allowed `SignIn` with `Apple`, `Google`, and `Email` paths explicitly.
- SRS-SEC-005: Cloud authorization-role enforcement is deferred from the current software package. If included in a future accepted baseline, authorization shall enforce least-privilege role and scope checks for telemetry and dashboard access.
- SRS-SEC-006: Protected-cloud authentication failure handling is deferred from the current software package. If included in a future accepted baseline, authentication and session failures shall fail closed for protected cloud operations while preserving safe local app behavior.
- SRS-SEC-007: Cognito password-recovery workflow is deferred from the current software package. If included in a future accepted baseline, email/password onboarding shall provide reset-code request and confirmation with explicit success/failure feedback.
- SRS-SEC-008: Launch session-restore behavior is deferred from the current software package. If included in a future accepted baseline, authenticated persisted sessions shall attempt secure restore/refresh without unnecessary manual re-login when restore succeeds.
- SRS-SEC-009: Auth-failure Home-bypass continuity behavior is deferred from the current software package. If included in a future accepted baseline, the app shall support explicit local Home bypass when unauthenticated local loop state remains active. While that bypass is active, Home shall keep a persistent top-level `Log In` control independent of alert ordering;

the login-required Home alert, Alert Center, Account & Session, and login-required notification shall provide direct login routes; and entering login shall not stop or reset the local algorithm session.

Team review notes:

- SRS-SEC-003 . . 009 are deferred from the current software package and require explicit scope re-entry before closure is claimed.
- Existing development implementation may cover portions of SRS-SEC-007 . . 009, but that implementation is not being claimed as closed scope in this package.
- Clinical, engineering, and administrative cloud roles and token-lifecycle controls are outside the accepted local-software scope and require explicit scope re-entry before they are claimed.

Traceability Source Mapping

Controlled traceability sources:

- this Software Requirements Specification;
- the Software Design Description;
- the Risk Analysis;
- the Software Verification and Validation Plan;
- the Requirements Traceability Matrix; and
- the controlled clinical-document consistency crosswalk in the IDE software packet.

Software Design Description (SDD)

Status: Frozen-baseline design description approved for IDE submission Version: 1.84 Owner: BionicLoop engineering Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, final software-package approval, controlled email dated 2026-09-07 EDT Baseline freeze SHA: 91c0e98a9bc9429a0486bebdebfffc7d8dbbe300e Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled design draft
0.9	2026-04-06	BionicLoop engineering	Added document-control metadata and disposition fields
0.91	2026-04-08	BionicLoop engineering	Added fallback-event review-state persistence for masked offline fallback
0.92	2026-04-08	BionicLoop engineering	Added the pump basal-schedule interface for masked offline fallback
0.93	2026-04-08	BionicLoop engineering	Clarified that programmed basal-schedule replacement is a disruptive masked-fallback arm/disarm operation that is blocked during active bolus, active temp basal, or non-steady basal-delivery states
0.94	2026-04-08	BionicLoop engineering	Clarified that masked-fallback schedule refresh uses a controlled reprogram-and-immediate-remask maintenance sequence and that unknown basal-delivery state blocks disruptive schedule replacement
0.95	2026-04-08	BionicLoop engineering	Added active masked-fallback maintenance lifecycle details, active-session gating, and reset/disarm restore behavior
0.96	2026-04-08	BionicLoop engineering	Clarified 20-minute masked-fallback renewal-window behavior, deferred maintenance semantics, and pre-step renewal before a new bolus-bearing step
0.97	2026-04-14	BionicLoop engineering	Added reconciliation-required blocked-state design notes after offline mask expiry
0.98	2026-04-14	BionicLoop engineering	Added reconnect restore/disarm retry notes and explicit fresh re-arm behavior after successful masked-fallback recovery
0.99	2026-04-14	BionicLoop engineering	Updated reconnect recovery design to preserve the existing session/cadence anchor and mark successful feasibility-path recovery as same-session unreconciled resume

Version	Date	Author	Summary of Changes
1.00	2026-04-15	BionicLoop engineering	Added basal-only reconnect evidence, pod total-delivery baseline capture, modeled-vs-pump-reported fallback review, and pre-step missing-fallback arm details
1.01	2026-04-15	BionicLoop engineering	Added confirmed/corrected basal-only reconnect replay design details, persisted replay-plan state, and replay-specific review detail in Home timeline surfaces
1.02	2026-04-15	BionicLoop engineering	Added dedicated cloud fallback telemetry emission from structured fallback review state for BionicScout timeline and dosing overlays
1.03	2026-04-15	BionicLoop engineering	Clarified that replayed internal steps now persist into local per-step telemetry / CSV export as distinct replay-marked rows
1.04	2026-04-27	BionicLoop engineering	Clarified that fallback replay uses pump-reported delivered-insulin delta only; modeled fallback exposure is logging/review evidence when pump reconciliation is unavailable
1.05	2026-05-06	BionicLoop engineering	Added Swift-maintained q5 nominal-basal profile persistence, safety-track four-bucket fallback schedule generation, schedule-aware exposure modeling, and profile/schedule cloud telemetry details
1.06	2026-06-02	BionicLoop engineering	Reframed masked-fallback recovery as pump-delta missed-step algorithm replay with unavailable CGM input, tightened pump-evidence credibility, restored IDE schedule-refresh cadence, and added repeated no-active-pod alerting
1.07	2026-06-08	BionicLoop engineering	Added persisted-schedule-weighted replay allocation for credible same-pod pump delta, including partial-slot / midnight weighting and zero-weight suppression rules
1.08	2026-06-12	BionicLoop engineering	Added generalized issued-dose attribution design for interrupted meal, correction, mixed, and basal bolus commands
1.09	2026-06-13	BionicLoop engineering	Hardened issued-dose attribution design for delivered-unit credibility, replacement-pod live-step input scrubbing, and automatic resume blocking under active holds
1.10	2026-06-15	BionicLoop engineering	Updated different/new-pod issued-dose design to assumed-delivered replay/live attribution, nonblocking replacement-pod dosing, legacy hold clearance, and deferred adaptation-forget consideration
1.11	2026-06-17	BionicLoop engineering	Added explicit meal-progress pod-replacement escape design with user-abandoned unavailable-pod evidence
1.12	2026-06-18	BionicLoop engineering	Added retired/expired/no-active-pod masked-fallback recovery design that records modeled fallback exposure as assumed delivered when the old pod cannot provide connected reconciliation evidence.

Version	Date	Author	Summary of Changes
1.13	2026-07-08	BionicLoop engineering	Added Algo2015 delivered-insulin echo persistence and cloud divergence telemetry for read-only insulin-conservation review.
1.14	2026-07-08	BionicLoop engineering	Added Algo2015 checkBG fingerstick prompt surfacing with haptic feedback, BG-button emphasis, dedupe, clear, and stale-relaunch rules.
1.15	2026-07-08	BionicLoop engineering	Added Algo2015 forced-open-loop command block and safety-critical alert surfacing while preserving fallback-basal maintenance.
1.16	2026-07-08	BionicLoop engineering	Added accepted no-boundary-cancel issued-dose liveness-bound design note for DASH delivery-state convergence and unavailable/unknown disconnect paths.
1.17	2026-07-08	BionicLoop engineering	Replaced the merged-first-fallback-row issued-dose design with pre-fallback confirm row + zero-delivery echo sentinels + seam ask echo (SDD-POL-029); no-room overlap keeps the merge with the requested side echoing the issued ask exactly.
1.18	2026-07-09	BionicLoop engineering	Exempted fallback replay backfill rows from interrupted-delivery chart styling (SDD-POL-027): offline asks are not pump commands, so their shortfall vs the pod-true share is not an interruption; issued-dose resolution rows keep the caution signal for genuine partial deliveries.
1.19	2026-07-09	BionicLoop engineering	Replay backfill bars render as a muted insulin-bar variant (whole-bucket/whole-pixel rule); documented the minimum-height zero-dose chart mark that distinguishes executed zero-dose steps (and sentinels) from missed steps.
1.20	2026-07-09	BionicLoop engineering	Scrub readout pills show only during active scrubbing with animated transitions; chart scrub state force-clears on view teardown and scene backgrounding; loop-status ring rendered without card chrome.
1.21	2026-07-09	BionicLoop engineering	Home visual polish pass: rounded monospaced hero BG value with baseline-aligned trend arrow, investigational-use chip, sliding-thumb chart range control, quieted solid-hairline chart grid with monospaced axis labels and CGM area fill, the primary actions share an outlined family (surface fill, accent-blue border) with chart-linked glyph colors - the compact Manual BG square carries a scheme-adaptive fingerstick-red droplet and Let's Eat carries the meal-dose purple fork - and the fingerstick-request state adds a faint caution wash under the amber ring so the glow reads anchored, 12 pt card-padding rhythm, and animated banner/section transitions. Display-only; IFU screenshots to be refreshed at next capture pass.
1.22	2026-07-09	BionicLoop engineering	Recent Dose Steps redesign: honest dose-settlement status policy (Delivered only with pump-anchored settlement; Requested for bare commands; Delivering/Assumed/Replayed),

Version	Date	Author	Summary of Changes
			compact summary rows with disclosure detail, and a clinical-unlock-gated CSV share action (SDD-LOG-001).
1.23	2026-07-09	BionicLoop engineering	Home alert stack with root-cause suppression matrix replaces the carousel (SDD-ALERT-001, SDD-POL-008); interrupted-dose chart bars keep identity color with a caution ghost outline to requested height and the partial-delivery summary wording is caution-colored (SDD-POL-027); step-drift preview catalog entry realigned to SRS-RUN-007 copy.
1.24	2026-07-09	BionicLoop engineering	Meal/BG modal cleanup: digit-keypad manual BG entry with inline range validation and single submit telemetry event (SDD-BG-001); meal carb choice as a fixed three-option row with unified subtle-fill selection styling, locale short-time availability copy, reserved composer heights with animated state swaps, neutral Cancel buttons, shared modal header, standardized sheet detents.
1.25	2026-07-09	BionicLoop engineering	Meal composer revalidation redesign (SDD-POL-004): in-place re-arm on stale observed step, inline self-healing blocked messaging with selections preserved, cancel-delivery routing retained; mid-composition dismissal into a second sheet removed.
1.26	2026-07-09	BionicLoop engineering	Home alert stack related-alert line (SDD-ALERT-001): hidden-count affordance into Alert Center when suppression is active; alert preview catalog extended with check-BG, dosing-paused (base + escalated), and backup-insulin-not-confirmed entries so the generated IFU reference covers every live alert; clinical-readability vocabulary pass on backup-basal event detail and algorithm-arm block messages.
1.27	2026-07-10	BionicLoop engineering	Added dual-instance algorithm hosting under SDD-ALG-001: the safety track binds a symbol-isolated second compilation so the two live instances share no C++ global state.
1.28	2026-07-10	BionicLoop engineering	Aligned forced-open descriptions with the implemented informational-first, escalation, and fingerstick behavior; added the fallback-replay-disposal acknowledgment; aligned alert, state-store, runtime-sequence, and requirement-allocation descriptions; and removed obsolete feasibility wording.
1.29	2026-07-14	BionicLoop engineering	Added SDD-OUTAGE-001 for fingerstick-supported CGM-outage dosing, including the persisted BG-due schedule, tiered alerts, Home countdown, backup-basal bridge, replay-based reconciliation, and no maximum-outage stop; added allocation rows for SRS-OUTAGE-001..006 and SRS-PUMP-011.

Version	Date	Author	Summary of Changes
1.30	2026-07-14	BionicLoop engineering	SDD-CLIN-001 extended for clinical-feedback changes: first-launch Standard/120/75%/40 defaults and the New Participant Reset design (full participant-scoped wipe + sign-out, armed/active-pod refusal, relabeled same-participant reset).
1.31	2026-07-14	BionicLoop engineering	Sponsor device-pass feedback: SDD-CLIN-001 updated for the grouped first-launch layout with per-mode target defaults, the defaults-without-persist empty-store load, the symmetric reset-button row, and the guided stop-then-reset flow.
1.32	2026-07-14	BionicLoop engineering	Aligned SDD-POL-029 with the implemented assumed-delivered policy; aligned the armed reset and sign-out paths; added the SRS-PUMP-011 watchdog design and RA-018 notification-permission limitation; and corrected the header date.
1.33	2026-07-16	BionicLoop engineering	SDD-ALERT-001 permission wording corrected to the actual mechanism (iOS drops delivery; the scheduler never skipped) and extended with the 2026-07-16 observability additions (authorization_status telemetry stamping + unauthorized-schedule cloud warning); no in-app logging controls remain (Settings cloud-log section removed; remote config per SRS-LOG-011).
1.34	2026-07-16	BionicLoop engineering	SDD-LOG-001 threshold-resolution tail corrected and extended: precedence stated as session > remote override > default error, the stale "local threshold exposed in debug builds via HomeSettingsView" sentence (invalidated by the 1.33 control removal) replaced with the read-only Settings status text design (CloudLogStatusSupport + HomeSettingsStudyLoggingSectionView), and the remote-config refresh triggers documented as launch/foreground/30-minute periodic (startPeriodicRefresh).
1.35	2026-07-16	BionicLoop engineering	SDD-LOG-001 tail re-stated for the SRS-LOG-010 supersession: session layer removed from precedence (server config > default only; retired session-key storage never read) and the configChangeObserver -> threshold-exempt remote_logging_config markers design added (bypass source list).
1.36	2026-07-16	BionicLoop engineering	SDD-APP-008 added: server-configurable investigational badge (CloudAppBannerConfigSupport three-state effectiveText, HomeTopOverlayBar TimelineView rendering) and the shared CloudRemoteConfigSupport.refreshAll/startPeriodicRefresh cadence for all server-issued configs (periodic loop relocated from CloudLogRemoteConfigSupport).
1.37	2026-07-16	BionicLoop engineering	SDD-LOG-001: G7 BLE trace emission follows the server-issued debug level when no local override is set; an explicit local off setting still takes precedence.

Version	Date	Author	Summary of Changes
1.38	2026-07-16	BionicLoop engineering	SDD-POL-029: meal-adaptation exclusion stated explicitly on every step cloud payload (meal_adaptation_exclusion_active true/false + request step/reason when active; sponsor request - excluded-or-not must never be inferred from silence; success path visible as the flag reverting without disposal evidence).
1.39	2026-07-16	BionicLoop engineering	SDD-OUTAGE-001: outage-surface gating aligned to the SRS-OUTAGE-002 v1.37 definition (field catch - relaunch blind steps no longer show a false fingerstick countdown).
1.40	2026-07-16	BionicLoop engineering	SDD-OUTAGE-001 added release-window quantification at re-arm, updating the engaged event with duration and modeled schedule exposure under RA-020.
1.41	2026-07-17	BionicLoop engineering	Extended SDD-CLIN-001 for the SRS-CLIN-014 telemetry drain guard: pre-confirmation flush, pending-count consent line, guarded asynchronous outbox discard with sequence continuity, and episode-latched operator alerting for permanent upload rejections.
1.42	2026-07-18	BionicLoop engineering	SDD-OUTAGE-001 aligned to SRS-OUTAGE-006 v1.42 (2026-07-18 field incident): released-window accounting restated as implemented live per-step crediting (OutageDeliveryCreditSupport echo + pod-delta tuples), pod-away gap reconciliation described as the coordinator-synthesized standard bounded replay (confirmed-stamped plan, echo chain, re-anchor + exactly-once accumulation), and re-arm quantification's pump-measured upgrade documented; the v1.40 interim-posture clause is closed.
1.43	2026-07-18	BionicLoop engineering	Aligned SDD-OUTAGE-001 with SRS-OUTAGE-006 v1.43 after independent safety review: 0.01 U quantized largest-remainder replay in the engaged profile timezone; persisted credit context across termination; and discard/re-synthesis after a cadence-drift skip.
1.44	2026-07-18	BionicLoop engineering	Documentation truth-sync: SDD-POL-029 and SDD-DATA-003 no longer treat active issued-dose reconciliation holds as an available or persisted mechanism. The writer-less hold was removed in c42f2a6; pending attribution, confirmed or assumed-delivered exactly-once consumption, discrepancy evidence, and lifecycle expiry remain the implemented controls. SDD-LOG-001 now records the sponsor-approved RA-009 local-file controls, including protection/backup exclusion for enumerated Algo2015 diagnostics.
1.45	2026-07-18	BionicLoop engineering	SDD-ALERT-001 check-BG presentation aligned to the study workflow: copy states that either a fingerstick BG check or sensor read can resume automated dosing; the alert has no inline Enter BG action, while the separate Home BG-entry control remains emphasized and available.

Version	Date	Author	Summary of Changes
1.46	2026-07-19	BionicLoop engineering	Exact-step pending-BG meal availability and pump-signal alert precedence: the forced-open meal block is bypassed only for a valid pending BG targeting the prospective meal step, with shared primary/safety consumption; pump signal loss ranks above same-severity check-BG while severity remains dominant.
1.47	2026-07-20	BionicLoop engineering	Manual-BG entry now reviews the frozen exact value in a native confirmation alert instead of mutating the submit button in place; direct check-BG, due-soon, and overdue requests share the Home gold breathing emphasis with a static Reduce Motion alternative.
1.48	2026-07-20	BionicLoop engineering	Manual-BG request emphasis now keeps the card shadow static and confines its repeating animation to a bounded gold stroke halo around the BG control, preventing chart-layer compositing artifacts while preserving the existing request policy and Reduce Motion behavior.
1.49	2026-07-20	BionicLoop engineering	SDD-POL-029 added the fresh-completion upgrade rule after a 2026-07-20 field incident: same-pod settled post-completion status can supersede deferred evidence with pump-observed completion, while canceled partials retain authoritative partial evidence.
1.50	2026-07-20	BionicLoop engineering	SDD-POL-004 (SRS-MEAL-006 v1.49): open-composer revalidation gains consumed-carve-out detection — carve-out context captured at composer open and on every in-place re-arm, and a <code>carveOutBGConsumed</code> flag that, only for the <code>backupBasalRunning</code> reason, swaps the generic backup-basal copy for the consumed-carve-out inline message; all other reasons and recovery routes unchanged.
1.51	2026-07-20	BionicLoop engineering	SDD-POL-004 corrected (SRS-MEAL-006 v1.50, sponsor correction of the v1.49 model): the consumed-carve-out flag becomes <code>carveOutBGConsumptionInFlight</code> (pending BG cleared while the captured target step has not yet published) and routes the <code>backupBasalRunning</code> reason to a <code>showTransientWaiting</code> hold — composer armed, passive applying notice, re-armed by the consuming step's publication — instead of any inline block or new-fingerstick demand; the same predicate guards backup-basal-blocked submits (pre-check and engine-blocked outcomes). Persisting forced-open after the consuming step published falls through to the generic backup-basal copy.
1.52	2026-07-21	BionicLoop engineering	SDD-LOG-001 amended (SRS-LOG-009 v1.51 steady-cadence fallback schedule visibility): <code>schedule_checked_unchanged</code> events attach the schedule + coverage payload minus <code>profile_rates_288</code> ; new <code>profile_observed_slot_counts_by_bucket</code> field on

Version	Date	Author	Summary of Changes
			arm/refresh/unchanged events (from <code>Q5NominalBasalProfileSchedule.observedSlotCountsByBucket</code>); Recent Dose Steps schedule rows add outcome label + per-segment observed/imputed provenance via <code>HomeRecentDoseStepsDisplaySupport</code> .
1.53	2026-07-21	BionicLoop engineering	SDD-POL-029 now persists explicit canceled-vs-in-progress-capped observation context and permits fresh completion upgrade only for positively identified in-progress caps; zero post-relaunch bolusNotDelivered no longer overrides canceled or legacy-unclassified partial evidence. SDD-ALERT-001 truth-synced to the accepted 569f17c centered shadow bloom (meal-selector envelope) and no-glow Reduce Motion treatment.
1.54	2026-07-21	BionicLoop engineering	SDD-POL-004 gains a shared prospective forced-open meal policy. <code>OutageBGRunContext</code> persists the algorithm step that consumed the last qualifying glucose; UI availability evaluates the exact prospective step, and the coordinator repeats the check after step resolution but before accepted-meal persistence. The characterized 12th blind step remains eligible; the 13th is rejected unless that step carries fresh CGM or a valid exact-step fingerstick.
1.55	2026-07-22	BionicLoop engineering	SDD-POL-004 meal-unavailable precedence corrected so a live connected <code>.delivering</code> status is presented as pump busy before pending attribution messaging; idle unresolved attribution behavior is unchanged.
1.56	2026-07-22	BionicLoop engineering	SDD-POL-004 consumed-BG revalidation now uses the exact step's persisted <code>lastQualifyingGlucoseStep</code> as its completion marker instead of <code>lastExecutedStep</code> alone, covering the pre-command maintenance checkpoint race. Home observes qualifying-glucose and pump-status changes to revalidate an open composer, re-arm on final BG publication, and replace stale backup-basal copy with connected pump-busy guidance when delivery starts.
1.57	2026-07-24	BionicLoop engineering	Added SDD-CLIN-002 and extended SDD-CLIN-001/LOG-001 for the Pregnancy Temporary Target lifecycle, execution-time primary-target resolution, permanent safety/fallback basis, explicit UI workflow, relaunch/expiry handling, Standard-to-Pregnancy meal default, and app-to-Scout observability.
1.58	2026-07-24	BionicLoop engineering	Hardened the SDD-CLIN-001/002 and SDD-LOG-001 New Participant Reset and Temporary Target telemetry boundary: reset gates new transport, waits a bounded interval for ordinary transport quiescence, durably erases before initiating the detached discard marker, rechecks live Pod/session guards, blocks concurrent algorithm arming, fully removes outgoing Temporary Target audit state, preserves legacy wire event IDs

Version	Date	Author	Summary of Changes
			while deduplicating them against the current deterministic UUID form, and pins execution telemetry to the configuration captured for that serialized work pass.
1.59	2026-07-30	BionicLoop engineering	Extended SDD-CLIN-002 with the reusable OmniBLE suspension control hosted directly on Exercise and Insulin Delivery, BionicLoop-only suppression of the duplicate Pod Settings Activity section, and one-shot suspend-alert navigation to the relocated control.
1.60	2026-07-30	BionicLoop engineering	Extended SDD-POL-016 with persistent signed-out Home login access, direct recovery actions from Alert Center and Account & Session, login-specific notification routing, and local-therapy/session continuity during authentication re-entry.
1.61	2026-07-30	BionicLoop engineering	Extended SDD-CLIN-002 with the reordered Settings entry, app-standard Temporary Target selectors/actions, removal of unrelated participant-reset guidance, and local draft-loss interception for target-only, duration-only, and complete unactivated selections.
1.62	2026-07-30	BionicLoop engineering	Extended SDD-POL-019 with the Home bottom safe-area action surface: Manual BG and Meal Announcement remain fixed while the reserved-inset content area scrolls, reusing the existing action handlers and emphasis policy.
1.63	2026-07-30	BionicLoop engineering	Replaced the episode-latched participant telemetry-rejection alert with durable failed-permanent outbox evidence (<code>failure timestamp + HTTP status</code>), preserved Scout gap detection and the specific Subject ID Conflict alert, retained 429 retry behavior, and added a one-time launch migration that clears earlier-build generic alerts/notifications.
1.64	2026-07-30	BionicLoop engineering	Extended SDD-CLIN-002/SDD-POL-008 with a direct, serialized OmniBLE resume command on the suspend-ended alert and persisted 15-minute actionable-to-safety-critical escalation with one-time re-notification.
1.65	2026-07-30	BionicLoop engineering	Extended SDD-POL-004/029 and Home status design so suspended delivery blocks meal execution before algorithm state advances, confirmed resume revalidates immediately, and definitively rejected bolus cache state cannot be reconstructed as later delivery feedback.
1.66	2026-07-30	BionicLoop engineering	Added SDD-CGM-002 for persistent explicit G7 replacement acquisition, raw-scan/replacement-scan state separation, timed in-app guidance, and durably deduplicated acquisition telemetry.
1.67	2026-07-31	BionicLoop engineering	Extended SDD-POL-022 with execution-outcome-based interruption diagnosis, explicit suspension/fallback-recovery blocker states, stale-blocker clearing after successful stepping,

Version	Date	Author	Summary of Changes
			and Home suppression of redundant interruption banners while the suspend-ended alert is active; aligned SDD-CLIN-001 reset teardown with asynchronous G7 persistence/telemetry isolation.
1.68	2026-08-06	BionicLoop engineering	Extended SDD-LOG-001 with persisted-input-derived fingerstick provenance in Recent Dose Steps, including exact manual BG value and concurrent CGM retention.
1.69	2026-08-11	BionicLoop engineering	Corrected SDD-ALG-001 host output mapping so each pump recommendation sums Algo2015 bolus, basal, and meal components exactly as the reference-host interface requires.
1.70	2026-08-11	BionicLoop engineering	Aligned SDD-CLIN-001 with the implemented initial-configuration boundary: no unlock while the persisted subject configuration is unusable/unset; the first successful save closes that exception and later access remains unlock gated.
1.71	2026-08-13	BionicLoop engineering	Extended SDD-LOG-001 with the v2 incremental telemetry spool, transactional v1 migration, retained-first recovery ordering, coalesced chatter-drop evidence, notification lifecycle deduplication, and passive upload-health presentation.
1.72	2026-08-17	BionicLoop engineering	Extended SDD-LOG-001 with the SQLite v3 retained-evidence ledger, async network drain/retry liveness, bounded backward-compatible batch upload, and a complete protected active-session step archive separate from the 576-row UI cache.
1.73	2026-08-20	BionicLoop engineering	Extended SDD-BG-001 with persisted pending-BG chart projection and explicit completed-step consumption evidence so the Home marker changes from outline to fill without timestamp movement or duplication.
1.74	2026-08-20	BionicLoop engineering	Extended SDD-LOG-001 with read-only discovery, epoch-session selection, bounded protected snapshots, stored-entry ZIP generation, and clinical-gated sharing for legacy Algo2015 artifacts.
1.75	2026-08-20	BionicLoop engineering	Corrected SDD-LOG-001 export presentation sequencing so session choice, confirmation, preparation, sharing, and failure remain within one stable sheet lifecycle instead of competing UIKit presentations.
1.76	2026-08-20	BionicLoop engineering	Replaced the unstable Settings export flow with a prebuilt combined CSV/Algo2015 recovery ZIP shared by the existing Recent Dose Steps <code>shareLink</code> ; exact epoch mismatch now falls back to CSV-only.
1.77	2026-08-20	BionicLoop engineering	Extended SDD-POL-022 with a status-refresh-only pump recovery transition: newer active-Pod idle evidence replaces stale unavailable copy under the existing alert key without waking the algorithm, and later unknown status restores the unavailable diagnosis.

Version	Date	Author	Summary of Changes
1.78	2026-08-20	BionicLoop engineering	Updated SDD-POL-019 to use point-only CGM chart rendering, removing the connector stroke and connected area fill while retaining dot styling, axes, and scrub behavior.
1.79	2026-08-21	BionicLoop engineering	Bound the controlled design description to the immutable 2026-08-21 software freeze and clarified the pending approval boundary. No design or product behavior changed.
1.80	2026-08-21	BionicLoop engineering	Added the candidate build-input control that requires CryptoSwift 1.10.0 exactly, tracks the app-workspace SwiftPM resolution, and fail-closed checks source/version/revision/Git control. No runtime, dosing, pump-command, or cryptographic behavior changed. The frozen baseline and archive 826 remain historical identities.
1.81	2026-08-27	BionicLoop engineering	Replaced internal development labels and test-device shorthand with reviewer-neutral descriptions, removed an obsolete unselected future-mitigation note, and updated the approval role. No design or product behavior changed.
1.82	2026-08-28	BionicLoop engineering	Completed allocation of the five active SRS identifiers omitted from the design-allocation table and reclassified the manual-BG wall-clock question as a deferred enhancement rather than an unresolved baseline decision. No design or product behavior changed.
1.83	2026-09-03	Software Developer	Recorded the frozen Build 843 G7 replacement-acquisition implementation limits found by the RA-023 adversarial review: direct manager reconstruction preserves an episode, but CGM-screen setup reentry can install a fresh manager; already-queued concurrent candidates lack an atomic single-winner gate; and the first accepted replacement reading can reach runtime before post-connection comparison. No product source or frozen design implementation changed.
1.84	2026-09-03	Software Developer	Aligned Build 843 configuration status, register references, alert-validation status, dependency wording, and submission-facing citations with the final software approvals and controlled appendix boundary. No product design or frozen implementation changed.

1. Design Intent

This design implements a CGM-triggered closed-loop runtime with:

- deterministic cadence state
- degraded execution behavior when inputs are unavailable
- explicit gating for pump command application
- traceable per-step telemetry

2. Major Components

- SDD-APP-001: `LoopRuntimeEngine` (`BionicLoop/Runtime/LoopRuntimeEngine.swift`) handles wake handling, availability logic, session control, and telemetry orchestration.
- SDD-APP-002: `LoopSessionStore` owns app-layer runtime persistence reads/writes for loop armed state and runtime state snapshots.
- SDD-APP-003: `LoopWorkScheduler` owns CGM-timestamp trigger dedupe and arm/reset semantics for runtime wake dispatch.
- SDD-APP-004: `LoopAlertMediator` owns pump signal-loss tracking and report gating in app-layer state.
- SDD-APP-005: `LoopTelemetryWriter` owns app-layer telemetry-store writes and reconciliation calls.
- SDD-APP-006: `LoopRuntimeWorkExecutor` owns coordinator `doWork` execution snapshot sequencing (latest reading capture, operation execution, completion timestamp, state snapshot).
- SDD-APP-007: `DeviceClockSyncMonitor` (`BionicLoop/App/AuthSessionNetworking.swift`) owns UTC midpoint drift checks (`GET /v1/time/utc`), retry policy, 24-hour foreground check gating, timezone/significant-time-change trigger handling, and lifecycle telemetry context projection.
- SDD-APP-008: `CloudAppBannerConfigSupport` (`BionicLoop/App/CloudAppBannerConfigSupport.swift`) owns the server-configurable investigational badge (SRS-UI-009): `parse ({ "config": { "text", "issued_by" } } envelope; missing/non-string text ignored; null clears)`, `UserDefaults` persistence, and the three-state `effectiveText` (absent config -> built-in NOT FOR HUMAN USE, configured text verbatim, explicit blank -> nil = no capsule). `HomeTopOverlayBar` renders the capsule from `effectiveText` on a one-minute `TimelineView` refresh. All server-issued configs share one fetch cadence via `CloudRemoteConfigSupport.refreshAll/startPeriodicRefresh` (launch, foreground, 30-minute loop) - the logging-config loop moved here from `CloudLogRemoteConfigSupport` when the second config was added.
- SDD-AUTH-001: App auth-session boundary is split into `AppAuthSessionManager` (secure token persistence, restore, refresh, sign-out clear) and `AuthenticatedAPIClient` (bearer-auth request composition + one-time refresh retry on unauthorized response), implemented in `BionicLoop/App/AuthSessionNetworking.swift`. Recovered/refreshed token sets are validated for telemetry-ingest scope before persistence.
- SDD-AUTH-002: Email/password recovery boundary is implemented in `CognitoPasswordRecoveryService` (`ForgotPassword`, `ConfirmForgotPassword`) plus unauthenticated route/state wiring (`.forgotPassword`) and `NoAuthForgotPasswordView` for reset-code request, confirmation, resend, and user-feedback messaging.
- SDD-CFG-001: Build 843 is the IDE submission build approved on 2026-09-03. Its controlled source requires `CryptoSwift 1.10.0` exactly in the `OmniBLE Xcode` project, tracks the app-workspace `SwiftPM` resolution, and applies a fail-closed source/version/revision/Git-control

check. This prevents an archive and its verification run from silently resolving different dependency versions. Archive identity, signing, installation, and evidence acceptance remain separate configuration controls.

- **SDD-CORE-001: LoopRuntimeCoordinator**
(`BionicLoopCore/.../Runtime/LoopRuntimeCoordinator.swift`) handles due-step logic, input assembly, algorithm invocation, and command application.
- **SDD-ALG-001: RealBUDosingAlgorithm + Algo2015Bridge** map runtime input into C bridge structs and receive output/state. The primary host constructs the pump recommendation as `bolusInsulinRequested + basalInsulinRequested + mealInsulinRequested`, matching the reference `AlgorithmInterface.h` and `AlgorithmController.deliverFluid` contract; the same total is persisted as the issued request and returned as pump feedback so `Reconcile_I_Dose` compares like-for-like values. Two live algorithm instances run per host step (primary dosing track; safety track computing raised-target fallback basal profiles): each binds its OWN symbol-isolated compilation of the unmodified algorithm + bridge sources (`Algo2015.xcframework / Algo2015Safety.xcframework`; the controlled safety build renames three extern "C" entries and localizes all other symbols, and its identity report proves byte-identical object code). This preserves the C++'s one-instance-per-process state model (file-scope globals, load-once) per instance; `RealBUDosingAlgorithm` selects the copy via `Algo2015EngineInstance` (2026-07-10).
- **SDD-PUMP-001: PumpServiceAdapter + PumpStatusObserver** handle status refresh, command execution, delivery reconciliation, home-status projection, and auto-polling while `deliveryState == delivering`. For the masked offline-fallback subsystem, the pump abstraction also carries a domain-level programmed basal-schedule seam (`PumpBasalSchedule`) so runtime can read and rewrite the pod's current basal schedule without taking a direct dependency on OmniBLE schedule types. On DASH, replacing the programmed basal schedule is a disruptive arm/disarm operation rather than a neutral setter: the OmniBLE implementation cancels current delivery before applying the new schedule, so the app seam now treats schedule replacement as an interrupting operation and blocks it unless pump delivery is in a known safe connected state (no active bolus, no nonzero temp basal, not suspended/resuming/suspending, and no unknown basal-delivery state). The accepted masked-fallback path uses that seam in two explicit maintenance operations only:
`performMaskedFallbackMaintenance(...)` replaces the programmed fallback schedule and immediately reapplies the `0.0 U/hr / 30 minute` mask, while
`performMaskedFallbackDisarm(...)` restores the original programmed schedule without reappling the mask. Runtime arms masked fallback immediately for a new active algorithm session once a valid fallback candidate is available from the current or latest successful step. If the candidate is first produced by the current step, `LoopRuntimeCoordinator` invokes a pre-pump-command maintenance hook after primary/secondary algorithm calculation but before applying that same step's pump recommendation; this lets step 0 program the fallback schedule and `0.0 U/hr` mask before a bolus command can place the pod into bolus-in-progress. The hook receives the same refreshed pump status used by pump-command

classification and skips fallback maintenance when that status is unavailable or unknown, so it cannot bypass the normal pump availability/unknown-state gate. If a valid candidate is already persisted but no fallback is currently armed, the existing pre-step maintenance pass first refreshes pump status and only arms fallback before the next loop execution when that status is idle; unavailable, unknown, delivering, or failed refresh leaves maintenance deferred without pod mutation. If the initial arm is cleanly blocked before any schedule mutation, runtime leaves fallback explicitly unarmed, records a failure instead of pretending ordinary deferred masked maintenance is active, and lets the current step's normal pump command proceed through its own safety gates; the same execution cycle does not then run post-execution masked-fallback maintenance for a second first-arm attempt. If the programmed fallback schedule may have been written but the immediate mask fails, is blocked, or is uncertain, `PumpServiceAdapter` normalizes that post-schedule mask error as `remaskFailed`, runtime persists reconciliation-required recovery context, and the current step's normal pump command is blocked before additional pod mutation. Runtime now maintains Swift-side q5 nominal-basal profiles from primary and secondary/safety algorithm scalar outputs; fallback programming prefers the secondary/safety profile, imputes missing 5-minute slots from observed values, averages four local-time six-hour buckets, rounds each bucket through the pump service, and programs the resulting four-entry `PumpBasalSchedule` (0, 21600, 43200, 64800 seconds) under the mask. If no q5 profile is available, runtime falls back to the latest rounded scalar safety-track nominal-basal candidate. Runtime refreshes the programmed fallback schedule no more often than the configured refresh interval when the desired schedule changes, preserves recoverable restore context on session reset/disarm until the original programmed schedule is successfully restored or explicit restore failure is recorded, and blocks the next algorithm start while unresolved restore-failed masked-fallback recovery remains in persisted state. For an already-masked fallback, steady-state connected behavior does not rewrite the mask every step: runtime tracks `maskExpiresAt`, enters a renewal window when 20 minutes or less remain on the active 30 minute mask, and performs pre-step renewal before the next loop execution only after a fresh idle pump-status refresh. If bolus, unknown/unavailable status, refresh failure, or other blocked pump state prevents renewal while the current mask is still active, runtime records deferred maintenance rather than immediate failure; `PumpServiceAdapter` now normalizes both device-state and late communication `unfinalizedBolus` outcomes on the temp-basal renewal path into that deferred-maintenance result while the current mask still has time remaining. If a remask failure occurs after an existing masked fallback renewal or schedule-refresh path and may leave fallback exposure uncertain, runtime records the exposure start (`maskExpiresAt` for renewal uncertainty, or current time for post-refresh remask failure) and enters reconciliation-required recovery rather than continuing ordinary stepping. If successful step execution stalls or deferral continues long enough for the active mask to expire, the pod is allowed to resume the programmed fallback schedule and runtime records that activation for later reconciliation rather than immediately remasking. Once `maskExpiredOffline` is recorded, runtime also persists a reconciliation-required recovery state and suppresses ordinary loop execution, manual BG dispatch, meal announce availability, and new arm until reconnect recovery resolves. While that state remains pending, runtime

attempts connected restore/disarm of the original programmed basal schedule on runtime start, app foreground, or pump reconnect. If restore succeeds while the loop remained armed, runtime clears the persisted masked-fallback state without resetting the current session, preserves the existing cadence anchor, and resolves basal-only reconnect evidence against the persisted fallback context. Exact cached-schedule equality is no longer a hard prerequisite for corrected recovery: if reconnect finds a different cached programmed schedule in local OmniBLE manager state, recovery may still record a `corrected` result and proceed with pump-delta reconciliation, but only when the persisted blackout context, proven same-pod continuity, a fresh connected pod-total-delivery baseline, and close agreement between modeled and pump-reported delivered insulin together provide credible basal-only evidence. App-log diagnostics capture both the persisted fallback schedule and the cached reconnect schedule from manager state for later review; that cached schedule is explicitly not treated as authoritative pod truth. Confirmed or corrected recovery persists pending pump-delta reconciliation metadata, records the resolved `disarmed` fallback event with reconciliation metadata, and lets `LoopRuntimeCoordinator` allocate the same-pod pump-reported delivered-insulin delta across missed 5-minute primary/secondary algorithm replay steps using the persisted programmed fallback schedule as the weighting model. Each missed replay step uses `CGM=-1`, that step's delivered-insulin allocation including explicit `0 U` input when no pump allocation belongs to that step, and no pump command application; the resumed live step then runs with actual refreshed pump status rather than a duplicate aggregate recovered-delivery input. If a positive pump delta cannot be weighted because persisted fallback schedule weights are unavailable or zero, runtime suppresses replay rather than fabricating an allocation; if the authoritative pump delta is `0 U`, the replay plan may still preserve the missed-step range so coordinator replay can emit `0 U` algorithm inputs. Pending replay metadata is single-use: if the live due step no longer matches, the plan is malformed, no session anchor exists, or the plan no longer contains missed steps before the current due step, coordinator emits a skipped replay trace and clears the pending plan instead of retrying it against later wakes. Ambiguous, different/new pod, unknown pod identity, or history-discontinuous recovery records a resolved `disarmed` event stamped as `resumed_without_reconciliation` and permits only the current due 5-minute step without replay. If the operator explicitly reset or turned the loop off first, successful recovery restores the original programmed schedule but keeps the loop off. The reconnect path now also exposes a basal-only `makeFallbackBasalReconciliationSnapshot(...)` seam carrying a stable pod identity, the cached programmed basal schedule from manager state, recent automatic temp-basal evidence, and pod cumulative delivered-insulin measurements so recovery can compare modeled fallback exposure against pump-reported delivered insulin since the last connected same-pod fallback baseline. Core now contains pure `FallbackBasalExposureReconciliation`, `FallbackBasalPodContinuity`, and `FallbackBasalReplayPlan` support types, and runtime/coordinator use them both for recovery classification/review and for no-command pump-delta missed-step algorithm replay when reconnect evidence is confirmed/corrected and same-pod; modeled exposure and replay weighting now integrate the persisted fallback schedule across six-hour bucket boundaries,

partial replay slots, and local-day midnight wrap rather than multiplying the entire blackout by one scalar rate. Current audible-command behavior remains split by the existing OmniBLE command categories: the automatic 0.0 U/hr mask temp basal follows automatic-command beep preference rules, while fallback schedule arm/refresh/disarm still reuse the manual/profile basal-schedule programming path and therefore may emit the normal DASH confidence beep unless pod confidence beeps are disabled or the pod is silenced. A silent maintenance-specific basal-schedule write path remains deferred backlog work. Every pump command and status await in `PumpServiceAdapter` is bounded by `PumpCommandWatchdog` (120 seconds): a timeout fails the command as uncertain (or the status read as blocked) instead of stalling the serialized loop work pass, and late or duplicate transport completions after the watchdog fires are ignored safely (SRS-PUMP-011); the pump-event storage delegate's synchronous-completion contract is regression-pinned.

- Retired/expired/no-active pod exception: when masked-fallback recovery is pending and `PumpStatusObserver` proves the previous pump session no longer has an active pod, or the observed time is beyond hard pod service stop, runtime may synthesize an assumed disarmed recovery event without connected schedule restore. The event uses `assumed_delivered_per_clinical_policy`, sets actual delivered fallback units equal to modeled schedule exposure for the bounded fallback-active interval, queues the normal no-command replay plan, clears the recovery state, and keeps the existing algorithm session. UI and telemetry label this as assumed delivered, not confirmed/corrected pump reconciliation.
- Replay eligibility is bounded to missed slots whose delivery interval overlaps confirmed fallback-active time. Missing slots that occurred after pump communication loss but before offline fallback activation remain absent from replay, not synthetic 0 U rows; explicit 0 U replay rows are preserved only inside the fallback-active interval.
- SDD-CGM-001: `G7ViewModel` + CGM subsystem adapters provide CGM state ingestion and wake trigger source; telemetry `cgm.state.changed` is emitted from refreshed/current callback state to avoid stale `has_sensor` transitions, and `cgm.connection.changed` derives `status_text` from refreshed lifecycle state before emit.
- SDD-CGM-002: `G7CGMManagerState` persists a replacement-acquisition episode separately from `G7Sensor.isScanning`: start time, prior sensor ID, trigger, the latest adoption event, and post-enqueue telemetry markers survive direct manager reconstruction.

`G7SettingsViewModel.scanningForReplacement` therefore controls a separate progress label without conflating ordinary BLE search for the bound sensor with user-directed replacement; the `Scan for new sensor` button remains visible and actionable during both states so an instructed retry can restart BLE discovery. `G7CGMManager.scanForNewSensor` records the episode, invokes a thread-safe synchronous app persistence handler with that snapshot, and only then directs the lower BLE layer to release its binding. Repeated triggers on the retained manager preserve the original clock/evidence, and authenticated glucose discovery records adoption and clears replacement state. Device-reported sensor failure/session end and the existing remote-disconnect-during-authentication heuristic can enter replacement mode; routine disconnect does not. `AppCGMManagerDelegate` derives the

ten-minute alert, owns and cancels its delayed evaluation task, retracts it on recovery, and emits stable-ID `cgm.state.changed` stall/adoption payloads from the common state-sync path independently of CGM screen lifetime, including the first sync after relaunch. An in-memory in-flight guard prevents duplicate telemetry enqueue; after durable outbox acceptance, the manager synchronously persists the matching marker, while failed enqueue remains eligible for a later retry. Stale reading age does not participate in replacement selection. Known frozen Build 843 limits are recorded under ANOM-007 and RES-010 in the IDE Known Anomalies and Residuals register, and D06 in the IDE Submission Decision Register: reopening CGM setup while the binding is cleared can install a fresh manager and reset the retained episode; already-queued candidate callbacks have no atomic single-winner adoption gate; and the first accepted replacement reading can reach runtime before staff complete the IFU post-connection comparison.

- SDD-BG-001: Manual BG entry path spans Home/UI entry, runtime `bgCheck` wake dispatch, single-pending-candidate BG state, next-eligible-step consumption logic, and telemetry source tagging (`manualBG`). The same runtime entrypoint now also checks persisted masked-fallback recovery state and refuses `bgCheck` dispatch while reconnect reconciliation is required. A valid pending BG may ride with meal input only when `pendingManualBGTargetStep` equals the exact prospective meal execution step; coordinator consumption supplies the same value once to both primary and safety algorithms and clears the persisted candidate. Once submission persistence succeeds, `LoopRuntimeEngine` publishes the stored pending value, submission time, and target step before running `doWork`, allowing Home to project an unfilled blood-drop marker immediately and after relaunch. Coordinator execution carries a `ManualBGConsumption` receipt through `DoWorkResult`; `LoopTelemetryStore` persists its original submission time beside the completed step's `AlgorithmInputSnapshot`. Home derives filled markers only from completed step records with valid positive `bgValueMgdL`, prefers that used evidence over a concurrent pending projection for the same target step, and retains the original submission time so the marker changes state without moving. Older step records decode without the optional timestamp and fall back to execution time. Pending-state disappearance alone is never interpreted as consumption. The entry sheet uses explicit digit-keypad entry (starts empty; a fingerstick value must be typed, never submitted from a stale prefill), a large rounded monospaced value display with inline range validation against 20...600 mg/dL (out-of-range shows safety-critical styling and disables submit), tightened no-CGM-calibration guidance, and a neutral (non-destructive) Cancel. Review BG freezes the valid entered value and presents a native, spatially separate Use `<value> mg/dL?` alert; Change dismisses the alert without clearing the digits, and only the alert's exact-value Use action submits. Rapid repeated taps on the entry action can therefore only present the review alert, not submit. A single UI-critical telemetry event carries the frozen final value at confirmation rather than one event per keypad tap.
- SDD-CLIN-001: Clinical settings boundary spans initial staff configuration, offline clinical-unlock-gated subsequent settings presentation, clinician-only control hosting (`Subject ID`, `Weight`, `Start Algo`, `Same-Participant Reset` - formerly `Reset Algo`), runtime-facing pregnancy configuration projection (`target`, `meal upfront`, `TMAX`), a clinician-controlled

participant target-access profile (Pregnancy / Standard), and the participant-facing target-change approval-capture sheet hosted in general settings.

`HomeClinicalSettingsSavePolicy.hasUsableClinicalSettings` defines the boundary: while subject ID or clinically plausible weight is absent, the initial review/save path is available without an unlock; the first successful save makes the configuration usable and all later Clinical Settings access/edits require the subject-scoped offline unlock. First-launch defaults are Standard/120 mg/dL/75%/40-minute TMAX (`HomeClinicalSettingsPolicy`, clinical direction 2026-07-14); defaults never rewrite persisted values. Clinical Settings additionally hosts the `New Participant Reset` (UI entry point

`LoopRuntimeEngine.performNewParticipantResetDiscardingTelemetryOutbox`, wrapping the synchronous `performNewParticipantReset`): a dual-confirmed, unlock-gated wipe of every participant-scoped store (config, algorithm/runtime state, Temporary Target state, local telemetry + CSV, cloud outbox, alert history, stored pump/CGM manager state, subject-scoped keychain unlock material, onboarding completion/claim), with sign-out performed by the app-root caller (`onParticipantSignOut` -> session store) after the engine wipe completes - `performNewParticipantReset` itself does not sign out. The reset is refused while a Pod is active; the pre-existing control is relabeled `Same-Participant Reset` and sits beside `New Participant Reset` in a symmetric two-button row; when a session is armed, `New Participant Reset` presents a guided `Stop Session` and `Continue` confirmation before the erase confirmation. Telemetry drain guard: tapping the reset (and again after a guided session stop) triggers a final `CloudTelemetryReporter.flushNow()` and re-reads `pendingUploadSummary()`; a nonzero pending count appends the discard-consent line to the erase confirmation via `HomeNewParticipantResetConfirmationPolicy` (nil count = no line until the drain settles). On confirm, the drain-guard checks the live armed/Pod guards, blocks concurrent algorithm arming, invalidates and drains queued runtime work, cancels Temporary Target drain/retry work, gates new outbox transport, and lets any ordinary transport already in flight quiesce for a bounded interval. Timeout or persistence failure returns `failedTelemetryErasure` and leaves participant stores intact. The engine rechecks live guards after asynchronous preparation and immediately before invoking the irreversible erase. `CloudTelemetryReporter.discardOutboxWithinNewParticipantResetTransaction()` then erases spool entries in every state while preserving the install-scoped sequence counter, reserves a sequence, and initiates the nonblocking best-effort `telemetry.outbox.discarded` marker; the telemetry gate remains held until the synchronous participant wipe completes, preventing outgoing-subject events from repopulating the spool. `AppCGMManagerDelegate.clearStoredParticipantState` cancels replacement-acquisition alert work, clears acquisition/adoption event IDs in flight, detaches both the ordinary manager-state delegate and the G7 replacement-state persistence callback, and releases its attached-manager reference before removing raw manager state; a late manager callback or cloud-enqueue completion therefore cannot repersist an outgoing participant's CGM state after reset. First-launch setup renders labeled groups (`Participant`,

Mode & Glucose Target, Insulin Settings) with per-mode target defaults and a review summary that includes the mode; `ClinicalAlgorithmConfigStore.load` returns first-launch defaults for an empty store WITHOUT persisting them.

- SDD-CLIN-002: `TemporaryTargetStore` owns a subject-bound, lock-serialized `UserDefaults` state containing the active override, pending lifecycle telemetry, first-applied evidence, and bounded effective-time intervals used to classify historical replay rows.

`LoopRuntimeEngine.activateTemporaryTarget` validates Pregnancy profile, armed-session, active-Pod, recovery-idle, explicit target, and explicit duration gates; activation persists before UI success, schedules an informational local expiry notification, and does not issue a Pod command.

`LoopRuntimeEngineOperationalSupport.primaryTargetResolver` resolves the primary target from each row's execution timestamp, while coordinator construction derives the safety target exclusively from `LoopConfig.permanentTargetMgd1`; Temporary Target therefore cannot change the safety instance, q5 profile basis, programmed fallback schedule, or session identity. Exact timestamp resolution makes the first row at/after expiry use the permanent target even after relaunch or missed notification. Replacement closes the prior interval and records a replacement lifecycle event before the new activation; early end, expiry, session stop/reset, profile change, and participant reset have distinct deterministic dispositions.

Session stop closes the active interval while preserving the historical interval needed for already-executed replay evidence; New Participant Reset cancels lifecycle drain/retry work and removes the complete outgoing-subject Temporary Target store without emitting or retaining a reset lifecycle record. `HomeTemporaryTargetView` provides the explicit no-preselection review/confirm flow on the combined Exercise and Insulin Delivery screen, while Home and Settings publish active target/end time. The Settings root places that destination before User; the screen uses the same outlined selected-state palette and primary/secondary action styles as the rest of the participant UI and omits the unrelated reset-location note.

`TemporaryTargetSelectionPolicy.hasUncommittedSelection` treats either selected field as a transient draft; while a draft exists, the view replaces native back navigation with an explicit discard alert and disables interactive sheet dismissal. Successful activation clears both draft values before ordinary navigation is restored. The same screen embeds

`OmniBLEInsulinSuspensionView`, a reusable presentation of the existing OmniBLE suspend/resume commands and four reminder choices; presentation policy disables commands for unusable Pods and transitional basal states while preserving the existing manual-resume requirement. The standard BionicLoop Pod Settings route passes `showsInsulinSuspensionControls = false` into `DashUICoordinator`, removing the duplicate Activity section without changing OmniBLE's default integration behavior. Suspend-alert notification routing creates a one-shot `HomeSettingsNavigationRequest` that presents Settings at Exercise and Insulin Delivery; other pump-alert routes remain unchanged. `HomeInsulinResumeActionModel` also exposes the same

`OmniBLEPumpManager.resumeDelivery` command directly from the active suspend-ended Home/Alert Center row, serializes taps while the command is in flight, surfaces a retryable communication error, refreshes pump status after command success, and deliberately leaves

alert retraction to pump-confirmed lifecycle evidence. The explicit Standard-to-Pregnancy draft transition is centralized in

`HomeClinicalSettingsPolicy.mealUpfrontPercent(afterTransitionFrom:to:currentValue:)`, leaving stored configurations unchanged until save.

`TemporaryTargetTelemetrySupport` emits subject-overridden durable lifecycle envelopes, and loop-step telemetry carries permanent/applied/safety target context; each serialized work pass retains the `LoopConfig` captured for that operation through lifecycle attribution and cloud emission, preventing a later configuration change from relabeling the completed pass. `BionicScout` merges lifecycle and step-use evidence by override ID and event time so delayed events cannot resurrect a terminal interval.

- **SDD-LOG-001:** `LoopTelemetryStore` keeps the newest 576 per-step records as the Recent Dose Steps presentation cache and independently upserts every current-session step into `SQLiteLoopStepArchive`; the protected clinical-gated CSV reads the complete archive, and explicit session reset clears it. `CloudTelemetryReporter` emits authenticated telemetry envelopes with correlation metadata (`event_id`, `session_id`, `subject_id`) plus `auth_user_sub` derived from Cognito ID-token sub (fallback `UNSET` only when unavailable); loop-step payloads carry explicit execution timestamp `step_executed_at` (execution semantics) in addition to envelope `created_at` (ingest creation time). Loop command telemetry mapping is schema-stable: `loop.command.blocked` is emitted only when a recommendation existed but was blocked, while `pump.command.result` remains pump-result schema only and emits on delivery-result deltas only (no unchanged refresh re-emission). For the masked offline-fallback subsystem, the same local telemetry store also persists structured fallback-basal review events published from runtime state so Home Recent Dose Steps can show fallback rate, source (`instant` vs `nominal`), safety target, duration, reconciliation status, pod-continuity result, restore/remask failure detail, modeled fallback delivery, and pump-reported delivered insulin since the last proven same-pod connected fallback baseline separately from ordinary step rows without adding blocking export work to the main actor. That local fallback event store now also acts as the app-side cloud emitter seam for a dedicated authenticated `loop.fallback.event` family: when a fallback event is newly created or materially updated, `LoopTelemetryStore` emits a structured payload carrying stable `fallback_event_id`, lifecycle kind, occurrence timestamps (`issued_at`, optional `reconciled_at`), selected/reconciled fallback rates, modeled delivery, pump-reported delivered insulin, `pod_continuity`, and reconciliation summary detail. Arm and schedule-refresh fallback events additionally attach the exact programmed schedule entries plus the safety q5 profile metadata used for that programming operation (`profile_source_track`, `profile_kind`, observed/imputed slot counts, the per-six-hour-bucket observed-slot counts `profile_observed_slot_counts_by_bucket` sourced from `Q5NominalBasalProfileSchedule.observedSlotCountsByBucket`, timezone fields, and `profile_rates_288`); the steady-cadence `schedule_checked_unchanged` check attaches that same schedule + coverage payload minus only the bulk `profile_rates_288` array (sponsor request 2026-07-20: a stable profile must not let the programmed fallback rates vanish from recent telemetry), and routine mask renewals attach no schedule/profile payload to

avoid high-volume telemetry churn. Recent Dose Steps fallback schedule rows (armed / schedule updated / schedule checked - unchanged) render a plain outcome label, the four programmed segment rates, profile coverage, and per-segment provenance; a segment with fewer than half of its 72 slots observed reads imputed; via `HomeRecentDoseStepsDisplaySupport` pure functions (`fallbackScheduleOutcomeText`, `fallbackScheduleProvenanceText`). Because a single logical fallback event can be updated later with reconciliation detail, the cloud contract treats `fallback_event_id` as the stable backend projection key while each emitted envelope keeps its own unique transport `event_id`. Executed step rows now retain both the primary algorithm telemetry snapshot and the secondary/safety algorithm telemetry snapshot plus each track's recommendation, so Recent Dose Steps can review both tracks' step count, suggested dose, nominal basal, and instant basal alongside the dose row. Recent Dose Steps rows use honest dose-settlement labeling via a testable status policy: `Delivered` is claimed only with positive pump-anchored settlement (idle refreshed status after an applied command, or replay resolution evidence), a bare command with no pod feedback reads `Requested`, active delivery reads `Delivering`, assumed clinical-policy evidence reads `Assumed`, fallback backfills read `Replayed`, and the requested-vs-delivered pair is shown only when a settled delivery genuinely differs from the ask (partials). Rows render as a compact summary (step, status chip, Meal/Replay/Discrepancy tags, dose, time) with the engineering detail - CGM tuple, algorithm input/output snapshots, pump evidence chain, primary/secondary track summaries, fallback schedule/profile/trace - behind per-row disclosure. The existing toolbar `ShareLink` exposes one protected snapshot ZIP containing the continuously maintained full-session step-telemetry CSV plus only the matching/current legacy Algo2015 session artifacts; it requires an active clinical unlock in release/study builds and is always available in DEBUG builds for engineering review. Algorithm output snapshots, CSV export, and loop-step cloud payloads also retain Algo2015's read-only delivered-insulin echo (`bolusInsulinDelivered`, `basalInsulinDelivered`, `mealInsulinDelivered`, and `deliveryTime`) and mark a cloud divergence flag when a matching pump-input delivery differs from the algorithm echo by more than pump mechanical resolution. When confirmed/corrected reconnect recovery occurs, `LoopRuntimeCoordinator` allocates the credible same-pod pump-reported fallback delivery delta into no-command missed-step primary/secondary algorithm replay rows with `CGM=-1` and schedule-weighted per-step pump-delivery input, then records the resumed live step with actual refreshed pump status and no duplicate recovered-delivery input. Reconnect diagnostics now also emit app-log evidence for every masked-fallback recovery attempt, including the persisted fallback schedule, the cached programmed schedule from local OmniBLE manager state, the original pre-fallback schedule, cached-schedule match state, cached-schedule source labeling, baseline/observed pod identity, pod-continuity result, modeled-vs-pump-reported delivered insulin totals, and no-replay reason when replay cannot be safely allocated, so backend review can determine whether schedule divergence reflects stale cached reconnect state, an incorrect schedule-programming command, zero/unavailable replay weights, or replacement/unknown pod continuity. The masked-fallback event lifecycle now includes explicit `armed`, `maintenance_deferred`, `mask_renewed`, `schedule_checked_unchanged`,

schedule_refreshed, mask_expired_offline, disarmed, remask_failed, and restore_failed transitions. Outbox flushing is loss-avoidant: durable enqueue completes before independent network work; concurrent flush requests coalesce with full-flush escalation, while scheduled retry work remains separate so a newer immediate drain cannot cancel an older row's retry. Client cloud-log threshold resolution uses precedence server-issued remote override (active, TTL-bound) -> default error threshold, persisted through CloudLogUploadPolicy; the server config is the sole study logging lever (the build-baked profile and the on-phone integration-session layer were removed 2026-07-16 pre-ship, and storage left under the retired session keys is never read). No in-app logging control is exposed in any build; Settings shows read-only effective-state status text (CloudLogStatusSupport mirrors the upload precedence exactly, rendered by HomeSettingsStudyLoggingSectionView on a one-minute display refresh). The remote override refreshes at launch, on each foreground, and on a nominal 30-minute periodic interval while running (the shared CloudRemoteConfigSupport.startPeriodicRefresh loop, SDD-APP-008); fetch failures leave the stored override untouched. A refresh that changes the stored override fires configChangeObserver, which the app wires to threshold-exempt remote_logging_config_applied/_cleared cloud markers (source bionicloop.remote_logging_config on the always-upload bypass list, so a cleared marker survives the very threshold drop it records). The G7 BLE trace (AppCGMManagerDelegate.emitBLETrace, subsystem BionicLoop.CGM/BLE.Core) follows the same lever: with no local env/defaults override set, trace emission is enabled exactly when the effective threshold is debug, so the server-issued debug level unlocks the full BLE narrative on study phones (an explicit local off still wins).

- Recent Dose Steps under SDD-LOG-001: HomeRecentDoseStepsDisplaySupport derives a Fingerstick tag and exact displayed value only from a positive persisted AlgorithmInputSnapshot.bgValueMgd1. When the same input also contains valid CGM, the display retains both values; it does not infer fingerstick use from wake cause or later UI state.
- Local-file controls under SDD-LOG-001: the CSV is written under Application Support with complete file protection and backup exclusion; the telemetry outbox uses a complete-until-first-unlock protected and backup-excluded file spool; participant builds omit Files sharing/open-in-place; and enumerated C++-created Algo2015 diagnostics receive complete-until-first-unlock protection plus backup exclusion. CloudTelemetryReporter classifies HTTP 4xx responses other than 429 as permanent, and CloudTelemetryOutbox persists the failed entry with its sequence number, event type, failure timestamp, HTTP status, and error detail. No generic participant alert is created; BionicScout detects the missing sequence, server logs retain the rejection, and a 409 subject-identity conflict independently raises only the specific actionable Subject ID Conflict alert. HTTP 429 remains pending for retry. AppAlertCenter performs a one-time launch cleanup for an orphaned retired telemetry-rejection notification; persisted-alert loading always filters the retired key and clears its scheduled notification so earlier-build state cannot survive upgrade.

- Incident-scale recovery under SDD-LOG-001: `SQLiteCloudTelemetryOutboxLedger` is the normal authoritative v3 store, configured with WAL, full synchronous commits, secure deletion, complete-until-first-unlock protection, and backup exclusion. It transactionally imports the prior file spool and monotonic counters, purges the source only after commit, and repeats idempotent import when a later transient SQLite failure caused the app to write a fallback spool. Retained clinical/runtime rows have no count cap; chatter rotates at 2,000. `CloudTelemetryOutbox.nextDeliverable` chooses retained evidence before chatter for an ordinary drain and preserves sequence order within each class. `CloudTelemetryReporter.emit` returns after the ledger commit, schedules independent coalesced drain work, and retains a separate earliest-retry timer. The optional batch path concatenates unchanged persisted envelope bytes into at most 50 events/512 KiB, applies per-event results, and returns inflight rows to pending before disabling batch on HTTP 404/405/501; the existing single route remains the compatibility path. Chatter cap evictions accumulate in durable metadata and produce one aggregate `telemetry.event.dropped` record only after an outbox delivery proves transport progress. New Participant Reset closes and physically recreates the outbox database, removes the database/WAL/shared-memory artifacts, and restores only install-scoped monotonic counters before reset completion; an erasure failure aborts the wider participant reset. The separate current-session step archive uses secure deletion and checkpoint/VACUUM cleanup when an explicit algorithm-session reset clears it. `UserNotificationAppAlertScheduler` persists active dedupe keys and emits clear telemetry once per scheduled-to-cleared transition across relaunch. Settings reads an atomic retained/chatter/permanent-failure summary and renders passive upload health; no participant alert or dosing dependency is introduced.
- Full-session export under SDD-LOG-001: archive reads, record decoding, and CSV construction run on a utility queue so archive growth cannot progressively block the main actor.
- Recovery export under SDD-LOG-001: `RecentDoseStepsView` prepares one archive off the main actor and supplies its URL to the pre-existing system `ShareLink`; no Settings export row, picker, confirmation alert, custom activity controller, or nested presentation remains. `LegacyAlgorithmArtifactCollector` requires the complete retained `BionicLoop_StepTelemetry.csv`, enumerates only the documented `BP_<epoch>.txt`, `BP_LOG_<epoch>.txt`, and `BP_ReadMatrix.m` patterns, and groups timestamped files by epoch while associating the fixed matrix only after parsing its referenced `BP_<epoch>.txt`. The latest algorithm inspection exposes the active C++ real-T0 epoch. An available epoch selects only an exact group; a missing exact group yields a CSV-only archive. If inspection state is unavailable after relaunch, the newest valid grouped epoch is used. Every included file is copied only to its size observed at copy start into a uniquely named complete-protected, backup-excluded temporary snapshot, then `LegacyAlgorithmStoredZIPWriter` writes stored entries so bytes are not decoded or transformed. The share action remains DEBUG-visible and otherwise clinical-unlock-gated.

The operation directory is removed when Recent Dose Steps closes. The algorithm writer, stream ownership, rotation, runtime execution, dosing, and telemetry persistence paths are unchanged.

- Confirmed/corrected reconnect replay telemetry is limited to fallback-active missed slots. Pre-activation disconnected gaps remain visible by absence rather than replay rows; 0 U replay rows are emitted only for fallback-active slots with no pump allocation.
- SDD-SIM-001: Deterministic simulation harness (medium-fidelity) uses protocol-conformant mock CGM/pump services plus a virtual clock/scenario runner to exercise runtime logic without BLE transport dependency.
- SDD-ALERT-001: Alert normalization/presentation layer is implemented as an app-level normalized model (`AppAlert* types + AppAlertCenter`) with deterministic precedence (`severity`, then condition-specific rank, then recency), dedupe keys, and debounced signal-loss handling. Active pump signal loss ranks above a same-severity algorithm check-BG prompt while every safety-critical alert remains dominant. Current live UI presentation includes a Home severity-prioritized alert stack (top alert always visible, compressed peek plus an expansion affordance for the rest, root-cause suppression per SDD-POL-008, and a tappable related-alert count line that opens Alert Center whenever suppression hides any active alert - `AppAlertPresentationPolicy.homeHiddenAlertCount`), plus a Home bell entry to Alert Center (active + recently cleared), with secondary access from Settings. Algo2015 `checkBG` output is synchronized after each successful algorithm output publication into a deduped actionable ALERT-ALGORITHM-CHECK-BG-REQUESTED prompt with foreground haptic feedback and Home BG-button emphasis. Its body states, "A fingerstick BG check or sensor read is needed for automated dosing to resume." The alert has no inline primary action; the separate Home BG-entry control remains emphasized and available.

`HomeManualBGRequestEmphasisPolicy` applies the same gold breathing emphasis to direct check-BG, outage due-soon, and outage overdue requests while excluding generic sensor, backup-basal, and pump alerts; Reduce Motion replaces breathing with a static emphasized ring. `HomeManualBGHighlightVisualPolicy` keeps the card shadow static and confines the repeating animation transaction to a bounded stroke halo around the BG control rather than the parent chart/actions hierarchy. This presentation-only policy reads existing active alerts and does not affect alert issuance, ordering, or runtime gates. The check-BG prompt clears on manual BG entry or a later `checkBG == 0`, suppresses repeated alerts while already active, and is filtered from restored active alerts at relaunch so stale requests do not survive without fresh algorithm output. Algo2015 primary or safety `openLoopMode == LOOP_MODE_FORCED_OPEN` output is synchronized into a deduped ALERT-ALGORITHM-FORCED-OPEN-LOOP prompt that presents informational-first (auto-clearing "Dosing Paused" note) and escalates to safety-critical advisory wording after 3 consecutive blocked steps, instructing a fingerstick BG to resume automated dosing plus sensor-check/study-team guidance (never a reset demand); it remains active across skipped/no-telemetry publications and clears after a later algorithm output reports forced-open inactive. The alert source inventory and mapping baseline is documented in Alert Inventory and Mapping. Pre-scheduled

OS notifications (stepping-interrupted, outage due-soon/overdue) depend on user-granted notification permission: authorization is primed once at app startup; scheduling always hands the request to iOS, which silently drops delivery when permission is denied or later revoked. Since 2026-07-16 the scheduler stamps `authorization_status` onto every `alert.notification.scheduled` telemetry event and logs a cloud warning when scheduling occurs unauthorized - observability, not user surfacing; the in-app permission-state indicator remains the identified in-app permission-status gap (RA-018 control gap), while the in-app alert stack and Home countdown remain permission-independent. Presentation clarification (v1.53; supersedes the stroke-halo details above):

`HomeManualBGHighlightVisualPolicy` keeps the ordinary card shadow static and applies a centered caution-amber breathing shadow bloom directly to the BG control using the meal-selector opacity/radius envelope. Reduce Motion disables that glow; the static amber border and wash remain the cue. This remains presentation-only and does not alter alert issuance, ordering, runtime, or dosing behavior.

- SDD-QA-001: Algo2015 verification infrastructure uses deterministic replay vectors plus instrumented coverage collection, mandatory static-analysis lane execution on formal runs, and conditional MISRA policy linkage (report+deviations or explicit N/A rationale) to generate reproducible TV-ALG-* evidence artifacts for IDE submission, as specified in Algo2015 Verification Plan.

3. Runtime Sequence (Nominal)

1. New CGM timestamp arrives.
2. `LoopRuntimeEngine` triggers coordinator `doWork`.
3. Coordinator computes expected step from anchor + interval.
4. Coordinator builds algorithm input:
 - CGM (or unavailable -1 by policy).
 - Pump fields (or unavailable values if pump not available).
5. Algorithm returns recommendation and updated internal state blob (primary track; the isolated safety-track instance computes fallback-basal recommendations through the same interface against its own engine and state store).
6. If pump status is available and recommendation passes gates, command is attempted.
7. Telemetry record is persisted with input/output/command fields.
8. Pump status refresh reconciles delivered/requested units.

4. Key Design Policies

- SDD-POL-001 (Cadence): anchored 5-minute slot model.
- SDD-POL-002 (CGM): step 0 strict gate, step >0 degraded allowed.
- SDD-POL-003 (Pump): unavailable pump permits algorithm step but blocks command application.

- SDD-POL-004 (Meal announce): explicit pump-ready gating (active pod present, known, and not delivering), established first-step cadence anchor requirement, borrow-window gating for pre-due execution, and current-due-step execution when request arrives after slot is due/missed. `OutageBGRunContext` stores the optional, decode-compatible `lastQualifyingGlucoseStep` whenever an executed step consumes positive CGM or fingerstick input. `MealAnnouncementOfflinePolicy` pins the real-CPP dropout boundary: at most 12 consecutive blind steps after that anchor remain eligible; a prospective 13th blind step is treated as the forced-open transition. Home availability computes the same prospective step as meal execution using the shared execution lead and blocks that transition before opening/submitting; a current fresh in-range CGM or valid persisted manual BG targeted to the exact prospective step permits the step. After exact meal-step resolution, `LoopRuntimeCoordinator` repeats the policy before `persistAcceptedMealExecutionState`; rejection returns `meal_announcement_would_enter_forced_open_loop`, performs no algorithm call or pump command, and maps to participant-facing `backupBasalRunning` copy stating that the meal was not announced and no meal insulin was delivered. Established forced-open meal availability remains blocked unless a valid persisted manual BG is targeted to the exact prospective meal step; invalid, stale, or future-target BG state cannot bypass the block, and all pump/reconciliation gates remain authoritative. Unavailable-reason precedence reports `noPump` before `signalLoss` when there is no active pod. After loop-off and masked-fallback reconciliation gates, an active pod with live `.delivering` status maps to `pumpDelivering` before pending meal/issued-dose attribution checks, preventing stale reconnect guidance while the connected pump is merely busy; pending attribution still blocks with reconciliation guidance once status is idle. A persisted masked-fallback reconciliation-required state blocks meal availability with reconnect/recovery-required guidance until that recovery resolves. Home/meal presentation logic revalidates meal availability on loop-step publication, qualifying-glucose publication, live pump-status change, foreground refresh, and immediately before submit. Revalidation of an open composer uses `HomeRuntimeActionCoordinator.mealComposerRevalidationDecision`: a slot conflict (observed step stale under the open sheet) with an available fresh check re-arms in place - observed step updated, selections preserved, a transient 'Timing updated' notice shown, telemetry result `recovered_in_place`; genuinely unavailable conditions render the current blocked messaging inline inside the composer (caution-tinted card, optional Open Settings action, selections preserved) and self-heal when a later revalidation reports availability; pump-delivering with an active cancellation context still routes to the cancel-delivery presentation. The composer captures the pending-BG carve-out context at open and on every in-place re-arm (`HomeRuntimeActionCoordinator.mealComposerCarveOutBGTargetStep` into `HomePresentationState.mealAnnouncementCarveOutBGTargetStep`); the revalidation decision takes a `carveOutBGConsumptionInFlight` flag. The helper requires a captured target, a cleared pending BG value, no matching-or-later `OutageBGRunContext.lastQualifyingGlucoseStep`, and no executed step beyond the captured target. This uses the final qualifying-glucose publication as the completion marker

because pre-command fallback maintenance may persist `lastExecutedStep == target` before app-level execution publication updates the outage context. Only `backupBasalRunning` routes to `showTransientWaiting` with the passive applying notice (`MealAnnouncementAvailabilityPresentation.carveOutBGApplyingNotice`); the composer stays armed and the qualifying-glucose publication re-arms it through the standard slot-conflict / inline-recovered routes, because an accepted fingerstick clears forced-open on its own step and restarts the full 60-minute dropout window (`Algo2015ManualBGDropoutClockCharacterizationTests`). A pump-status change revalidates the same composer so connected `.delivering` guidance supersedes any intermediate waiting or backup-basal copy immediately. The same predicate (`mealComposerSubmitShouldHoldForCarveOutBGConsumption`) holds a backup-basal-blocked submit on the waiting notice at both the pre-check and the engine-blocked outcome instead of tearing down to the unavailable sheet. Once the qualifying marker publishes, or a later executed step proves the target passed, persisting forced-open availability keeps the generic backup-basal copy; all other unavailable reasons and every recovery route are unaffected by the flag. This is the accepted current software baseline for late-submit meal behavior. Suspension hardening (v1.65): both Home availability and `MealExecutionPolicy` require the active Pod's delivery state to be `.idle`; `.suspended` maps to `pumpSuspended` with resume-before-meal guidance. The coordinator rejects before accepted-meal persistence and before either algorithm runs, so a suspended meal cannot advance cadence, produce a 2.9 U-style recommendation, issue a command, or seed chart/feedback state. `LoopRuntimeEngine` evaluates the freshest service status ahead of observer fallback, so a pump-confirmed `.idle` status after Resume makes meal entry available immediately without waiting for another loop step. Home separately renders `Insulin delivery is paused.` from live pump status, and the pinned action surface has no decorative top hairline.

- SDD-POL-005 (Reset): full session reset clears persisted runtime + algorithm + timeline session data.
- SDD-POL-006 (Home status): state precedence is deterministic and runtime-derived; age-based states use cadence phase (`nextDueAt` from anchor + step index) so borrowed meal steps do not falsely age status before the next due boundary. Current thresholds: Active through `nextDueAt + 2m`, Aging through `nextDueAt + 15m`, then Stale.
- SDD-POL-007 (Modal setup cancellation): startup/setup modals for CGM and Pod provide direct dismiss behavior via explicit `Cancel`, including persisted-manager/no-active-pod Pod setup.
- SDD-POL-009 (Meal composer lifecycle safety): if app scene transitions to background while meal composer is presented, the composer is auto-cancelled/dismissed.
- SDD-POL-010 (BG check): manual BG submit triggers `bgCheck` with no borrowing, uses a single pending BG candidate (replace-on-new submit), applies candidate to the next eligible step, and expires candidate if not consumed on that immediate next step.
- SDD-POL-011 (BG safety gate): manual BG submit path checks loop-armed state plus persisted masked-fallback recovery state in `LoopRuntimeEngine` and shall not dispatch `bgCheck` while disarmed or while reconnect reconciliation is required.

- SDD-POL-012 (BG step-0 guard): until step-0 BG rescue is approved, manual BG submit is rejected if `firstSuccessfulStepAt` is not established.
- SDD-POL-013 (Clinical settings policy): clinician configuration access is gated by `ClinicalUnlockVerifier` and `KeychainClinicalUnlockStateStore`. The app verifies subject-scoped single-use unlock codes offline after verifier material is provisioned, stores verifier material/local state per subject in the Keychain, burns the matched counter before exposing clinician controls, and keeps the clinician controls unlocked only until the provisioned expiration timestamp or explicit manual lock. The app-side material provisioning path validates the verifier material before storage, rejects subject/material mismatches, preserves the last accepted counter when material is refreshed, clears active unlock expiration on refresh, and resets local failed-attempt/lockout state so refreshed material does not silently preserve an open clinician session. The verifier normalizes pasted/grouped ASCII numeric input, computes the HMAC code using the current provisioned secret version, searches only the configured future counter lookahead, and rejects malformed, non-ASCII digit, reused/lower-counter, wrong-subject, beyond-lookahead, and unsupported-version codes. Manual lock writes the cleared unlock expiration back to secure storage; if storage persistence fails, `HomeSettingsView` keeps clinician controls visibly unlocked and shows the storage-failure message instead of presenting a false locked state. Production master secrets are not embedded in app source; DEBUG UI-test provisioning uses the public contract test secret only under `UI_TEST_MODE` and exercises the same app-side provisioning path. Operational provisioning is human-typed: `ClinicalUnlockProvisioningCode` (BionicLoopCore) decodes the BionicScout-issued one-time provisioning code (Crockford base32 of a 30-byte payload carrying format version, secret version, code digits, offline lookahead, unlock duration, a 4-byte subject binding check, the 16-byte per-subject derived device secret, and a 2-byte checksum; input normalization maps I/L to 1 and 0 to 0 and strips separators), and `ClinicalUnlockProvisioningInstaller` installs the decoded material through `ClinicalUnlockSessionController.provision`. `HomeClinicalSettingsFormView` presents provisioning as an operator-facing "Clinical Unlock Setup" section only while clinician controls are locked: when material is installed for the current subject it collapses to a ready confirmation (subject + code-set version, with an explicit "Redo Device Setup" action); otherwise it shows a guided form with "Set Up Automatically" (authenticated fetch) as the primary action and the typed setup code behind a manual-entry disclosure, and the locked unlock-code field shows a set-up-required hint. On successful installation `HomeSettingsView` adopts the provisioned subject ID only when the app has no subject configured yet (bootstrap) and never silently changes a non-empty configured subject. `ClinicalUnlockAutoProvisioner` performs the fetch through `AuthenticatedAPIClient` (`GET /v1/subjects/{subject_id}/clinical-unlock-material`, subject-device token with automatic refresh), maps auth/claim failures to operator-readable messages (`notSignedIn`, `subjectNotClaimed`), rejects responses whose `subject_id` differs from the requested subject, and installs the fetched provisioning code through the identical `ClinicalUnlockProvisioningInstaller` path so typed and fetched material share one validation surface. The device stores only the per-subject derived secret, so compromise of

one device's Keychain material exposes only that subject's unlock codes. Clinical control values are validated to allowed ranges (target 90...130 step 10, meal upfront 75/90, TMAX 40...70 step 5) and mapped into algorithm/runtime configuration without changing existing start/reset semantics. `HomeSettingsView` persists a target-access profile (Pregnancy, Standard) alongside the clinical config; participant-facing settings derive their available target buttons from that profile (Pregnancy → 90/100/110, Standard → 110/120/130). The clinician target selector now uses that same profile subset and normalizes any inherited out-of-profile draft target to the nearest allowed value when the profile changes. When a previously conflicting subject identifier is corrected and the save/claim flow succeeds, `HomeSettingsView` explicitly retracts the app-level `ALERT-SUBJECT-ID-CONFLICT` alert so operator-facing alert state matches the resolved persisted configuration. Separately, Home now performs a throttled silent re-claim check for the currently persisted subject identifier when the conflict alert is still active and the operator launches, foregrounds, or reopens settings; if the backend claim now succeeds, Home retracts the stale conflict alert without requiring a settings edit/resave round-trip. This investigational single-device unlock control remains distinct from broader production role-based authorization.

- SDD-POL-014 (CGM display masking for stale/unreliable data): `G7ViewModel` applies a display-staleness cutoff of 11 minutes and checks `hasReliableGlucose` on the latest G7 reading. If stale or unreliable, display accessors emit masked glucose text (`-- mg/dL / --`), suppress trend-arrow rendering, and do not append unreliable points into chart-history persistence.
- SDD-POL-015 (Auth/runtime decoupling): authentication/session state shall not stop or reset local loop runtime state. Auth gates cloud/protected operations and onboarding surfaces, while therapy runtime continuity is preserved unless user explicitly performs runtime reset/start-stop actions.
- SDD-POL-016 (Auth re-entry with continuity fallback): app launch attempts silent token-session restore/refresh when persisted auth state is authenticated; on restore success, authenticated UX remains uninterrupted. If auth state changes to unauthenticated while restore is in flight (for example explicit user sign-out), launch reconciliation preserves the current unauthenticated state and does not auto-reauthenticate. When restore fails (or user is unauthenticated) and local runtime indicates active algorithm state, login UX exposes explicit Home bypass. In bypass mode, `HomeTopOverlayBar` keeps a persistent `Log In` control visible independent of active-alert precedence; the Home alert, Alert Center active row, and signed-out Account & Session screen expose the same recovery action. `ALERT-AUTH-LOGIN-REQUIRED` notification taps route to the login destination instead of generic Home. Every route calls the app-level login presentation handler, which closes Home modals and displays authentication without stopping or resetting the local algorithm session.
- SDD-POL-017 (Clinical/settings telemetry): Clinical Settings save-review flow emits `ui.critical` telemetry with stable event/action mapping: `state_viewed` on review presentation, `submit` on Save, `cancel` on review dismissal, and `blocked` on invalid/locked/no-change save attempts. Save telemetry includes changed-field list and old/new clinical values (`subject`, `weight_lbs`, `target_mgdl`, `target_range_profile`,

`meal_upfront_percent`, `tmax_minutes`). Participant target-change approval capture emits its own stable `ui.critical` lifecycle (`state_viewed`, `submit`, `cancel`, `blocked`) with requested target, current target, target-access profile, approving staff name, and approval timestamp.

- SDD-POL-018 (Clock-sync safety telemetry): App lifecycle telemetry (`app.lifecycle.launch`, `app.lifecycle.foreground`) includes `timezone` + UTC-check context fields (`device_timezone_id`, `device_utc_offset_seconds`, `clock_check_result`, optional skew/rtt/timestamp). UTC checks run on launch, `timezone/significant-time-change` notifications, and foreground only when last successful check is older than 24 hours. If absolute skew exceeds 600 seconds, app raises actionable `ALERT-APP-CLOCK-SKEW` warning no more than once per 24 hours; unavailable checks do not show user warnings.
- SDD-POL-019 (Home chart and primary-action layout): CGM value rendering maps ≤ 39 mg/dL to LOW and ≥ 401 mg/dL to HIGH across Home/G7 primary value surfaces and scrub value presentation; Home inline CGM chart uses bounded stepped maxima (300, 350, 400 mg/dL) based on observed peak values so out-of-range excursions remain visible without unbounded axis drift. `InlineCGMChartView` renders each glucose sample as one value-colored dot at its recorded timestamp and does not render the former connector stroke or connected area fill, avoiding visual interpolation between discrete sensor samples. Scrub readouts (CGM/Dose value pills above the combined Home chart) appear only while a scrub gesture is active, with animated show/hide transitions, and every chart scrub state is force-cleared on view teardown and on scene backgrounding - a cancelled drag never fires the gesture's end handler, so lifecycle clearing guarantees scrub overlays cannot stay stuck. The Home loop-status ring is a passive indicator rendered without card chrome (it is not a button). `HomeView` places the existing `HomePrimaryActionsRow` in a bottom `safeAreaInset`, with an opaque background and top divider; SwiftUI reserves that inset from the scrollable status/alert/chart region, so Manual BG and Meal Announcement stay visible without covering clinical content. The fixed row reuses the same BG-request emphasis and action handlers, so placement does not change availability, meal preflight, BG validation, or dosing behavior.
- SDD-POL-020 (algorithm stepping interruption monitoring): `LoopRuntimeEngine` monitors for stalled loop execution while armed using persisted runtime/session state rather than CGM-receipt time alone. The interruption deadline is computed from `lastSuccessfulRunAt` + 15 minutes; if no successful step exists yet, the baseline is session start / first-step wait state. `AppAlertCenter` clears any prior pending interruption notification and schedules a replacement local-notification deadline at the current interruption deadline; if the deadline passes while still current, it raises actionable in-app alert state. Monitoring is recomputed on app foreground and refreshed after each successful step. The alert clears when successful stepping resumes or when the loop is disarmed/reset.
- SDD-POL-021 (reconnect fallback execution): current cadence remains anchored to `firstSuccessfulStepAt` with the existing early execution lead (27 seconds). Pump reconnect is treated as a secondary wake source rather than a cadence source: app runtime listens for pump-status recovery (not raw BLE connect), only after an anchored session exists,

only for step >0, only when the accepted CGM receipt is older than the approved freshness limit (>5 minutes), and only when the current due slot has not already executed. The reconnect path may execute only the current due step, shall not replay missed slots, shall not re-anchor the schedule, and shall remain suppressed once fresh CGM receipts resume so CGM retains priority as the primary loop trigger.

- SDD-POL-022 (algorithm stepping interrupted alert policy): the normalized runtime interruption alert is `ALERT-ALGORITHM-STEPPING-INTERRUPTED`, keyed to lack of successful algorithm execution rather than lack of CGM receipt alone. Trigger basis is `loop armed` plus no successful step for >15 minutes, measured from the most recent successful executed step or, if none exists yet, from session arm time / first-step wait state. The alert issues even when the immediate blocker is not CGM-specific.

`LoopRuntimeEngineStepInterruptionMonitoringSupport` classifies the latest unresolved `doWork` skip plus authoritative pump-native conditions into blocker-specific detail; suspension and fallback reconciliation have explicit states, broad CGM lifecycle alerts do not independently establish step causality, and current suspension outranks fallback recovery while fallback recovery outranks an older persisted CGM/pump-status diagnosis. Following a `pumpStatusUnavailable` skip, the existing pump-status observer refresh path reevaluates interruption copy without calling `doWork`: only an active Pod status with `deliveryState == idle` and `lastUpdated` later than the failed attempt enters the persisted `pumpRecoveredAwaitingNextStep` state and changes the same deduped alert to `communication-restored/waiting-for-glucose` guidance. Status-driven refresh applies the alert update only when the persisted blocker changes, so repeated newer idle polls do not churn alert timestamps, lifecycle telemetry, or notification attempts. Stale or non-idle status remains unavailable; a later unknown status returns the persisted diagnosis to unavailable. The recovered-waiting state survives relaunch until a newer result or contradictory status supersedes it. Any successful live step clears interruption state. `AppAlertCenter` updates changing blocker copy under one dedupe key; Home hides the redundant interruption banner while the stronger suspend-ended alert is active, while Alert Center and telemetry retain the interruption record.

- SDD-POL-023 (meal-request durability baseline): `LoopRuntimeState` persists pending meal-request state across relaunch using submit time, the concrete execution step accepted by `LoopRuntimeCoordinator`, a pending-state discriminator (`awaitingExecution`, `uncertainCommandOutcome`), and the correlated meal flow identifier. On startup/foreground availability checks, runtime reconciles that state against persisted step history (`lastExecutedStep`) and, for uncertain outcomes, against pump-delivery evidence (`lastDelivery.requestTimeStep`) after pump attachment. While the pending request remains unresolved, `mealAnnouncementAvailability()` returns `requestInProgress` or `uncertainPreviousMeal` and duplicate meal entry is blocked until reconciliation or session reset/disarm.
- SDD-POL-024 (meal outcome handling): Home/UI and runtime telemetry split meal flow into pre-submit interaction plus outcome-driven `submitted`, `accepted`, `success`, `blocked`, `uncertain`, and `resolved` paths. `HomeView` keeps the composer in a submitting state while

`LoopRuntimeEngine.announceMeal()` awaits coordinator completion, dismisses only on executed success, presents explicit blocked messaging when runtime returns a blocked/rejected outcome, and presents explicit uncertain-delivery guidance when pump command outcome is unresolved. Home captures the `lastExecutedStep` observed when the meal composer opens and passes that snapshot with submit; `LoopRuntimeCoordinator` compares it against current cadence state and returns `mealSlotConflict` if another step has already advanced, so runtime maps the result to explicit slot-conflict retry guidance instead of silently reassigning the meal into a new borrowed step. Runtime now emits correlated accepted telemetry after coordinator acceptance and resolved telemetry when the request clears immediately, reconciles after uncertainty, or is cleared by session reset, using persisted replay state for the meal flow identifier + target step until those lifecycle events are emitted so cloud review can close the lifecycle across relaunch/terminate windows. Loop-command telemetry carries explicit `command_outcome` semantics so uncertain delivery is not flattened into generic blocked state.

- SDD-POL-025 (participant target-change approval flow): The general settings target control is a constrained participant-facing action, not a direct free-form clinical config editor. `HomeRegularTargetChangeApprovalPolicy` first validates that the requested target belongs to the clinician-selected profile subset and that the request would change the applied target. Valid requests open an approval-capture sheet that blocks dismissal until the operator either cancels or records both approving staff name and approximate approval time. Only after successful validation does the app persist the updated target into the shared clinical config store and update the live runtime-facing target binding. Invalid approval-capture attempts emit blocked telemetry and keep the prior target unchanged.
- SDD-POL-026 (CGM urgent-low review alert): `AppCGMManagerDelegate.syncG7Alerts(for:)` derives `ALERT-CGM-URGENT-LOW` from current live G7 data when `latestReading.hasReliableGlucose == true`, the reading timestamp is within the trusted freshness window, and `glucose < 55 mg/dL`. The alert is app-derived for clinician review and cloud telemetry, not Dexcom-app alarm acknowledgement. `AppAlertCenter` keeps the alert active after acknowledge by stamping `acknowledgedAt` rather than clearing it immediately, so Home / Alert Center can display reviewed state while the urgent-low condition remains active. The active urgent-low alert auto-clears only when a later trustworthy reading is `>=55 mg/dL`; stale, unreliable, or missing readings leave the active episode in place. Repeated live-state upserts preserve existing `acknowledgedAt` for the active episode, and a later new `<55 mg/dL` episode reissues as unacknowledged after recovery clears the prior episode. `CGMAlertPersistenceStore` persists acknowledged dedupe state so reviewed active urgent-low episodes survive app reset / delegate reattach until glucose recovery clears the episode. As with all CGM-derived alerts, this path never schedules OS local notifications.
- SDD-POL-027 (meal cancel-delivery path): When the operator opens meal announce while pump bolus delivery is already in progress, `HomeActionCoordinator` routes the flow into a dedicated cancel-delivery presentation on Home instead of exposing a second meal submit path. `HomeView` reveals a destructive slider-backed `Cancel Delivery` action beneath the

Manual BG / Let 's Eat controls, and that action can now remain visible automatically while an active meal-announcement bolus is still in progress, even without a second tap on Let 's Eat, by deriving the active cancelable state from current pump delivery state plus the matching meal step telemetry. When `PumpStatus.lastDelivery` is unavailable during reconnect/status-thin windows, Home falls back to the latest still-delivering meal step record so the cancel affordance is not hidden solely because detailed delivery accounting has not yet arrived. The cancel action calls `PumpStatusObserver.cancelBolusDelivery()`, which delegates to `PumpService.cancelBolus()` and converts the canceled dose into actual requested/delivered units. On success, Home shows a caution-colored partial-delivery summary in the normal alert-summary region above the chart so the operator can see how much insulin was already delivered before cancellation before later reopening meal announce. That summary now carries cancel time plus meal type/size context derived from the matching meal step telemetry, so the operator can distinguish which meal was interrupted without reopening the composer. The summary is retained until both at least one later algorithm step has executed and at least 5 minutes have elapsed, preventing the warning from disappearing immediately after the next step. The Home insulin chart derives interrupted-delivery styling directly from existing step telemetry (`requestedUnits` vs `deliveredUnits`), but active in-progress meal delivery remains rendered in the normal meal-dose color while the pump still reports `.delivering`; `LoopTelemetryStore` now marks a successfully issued bolus as delivering immediately until the first pump refresh arrives, so a just-started meal bolus does not flash yellow while waiting for hardware status. Interrupted/partial deliveries keep their data-identity bar color (meal purple / insulin blue) at the delivered height, with the caution treatment rendered as a ghost outline up to the requested height - the bar reads "what was delivered vs what was intended" without erasing what kind of dose it was; the caution ghost appears only once delivery has been stopped or otherwise ended short. Fallback replay backfill rows (fallback-only replay rows with no issued-dose evidence source) are exempt from interrupted-delivery styling: they pair the algorithm's offline ask with the pod-true quantized fallback share and issued no pump command, so a requested-vs-delivered shortfall there is historical reconciliation, not an interruption. Replay rows carrying an issued-dose resolution report a real command outcome and keep the caution treatment for genuinely partial deliveries. Fallback replay backfill rows additionally render as a muted variant of the insulin bar color (reduced-opacity tiers at every scrub state) so historical reconstruction reads differently from live-commanded doses; the muting applies only when an entire display bucket/pixel is backfill - any live dose in the same bucket keeps normal styling. Every executed step renders a minimum-height chart mark even at 0 U delivered (including dead-gap sentinel rows), distinguishing 'the loop ran here with a zero dose' from a genuinely missed step, which has no record and renders nothing. When cancel succeeds, `PumpStatusObserver` explicitly reconciles the shared `LoopTelemetryStore` step record with the cancel-returned delivered amount and forces the corresponding local delivery state back to idle, so the chart bar height reflects actual delivered insulin even if the immediate post-cancel status refresh or lagging `LoopKit` event history still overstates the completed amount. Later pump-status observer refreshes continue reconciling the shared `LoopTelemetryStore` record for the same step until delivery completes, so an in-

progress partial snapshot does not remain yellow on the chart after a later refresh proves full delivery completed. The chart render compactor preserves `.delivering` when adjacent bars collapse onto one display pixel so compaction cannot strip active-delivery styling. The pump adapter preserves the delivered amount in `PumpStatus.lastDelivery` so the next algorithm step consumes actual partial-delivery evidence rather than assuming the full original bolus completed, and when LoopKit event history lags a later pod status refresh it now floors delivered insulin to the pod-reported `requested - bolusNotDelivered` amount instead of regressing a completed bolus back into partial-delivery state.

- SDD-POL-028 (operator-controlled algorithm session lifecycle): `LoopRuntimeEngine` treats algorithm start, stop, reset, and replacement session creation as explicit operator actions only. Recovery code paths may preserve or clear blocked fallback state, retry pump schedule restoration, suppress replay, continue the existing cadence slot, and emit unreconciled telemetry, but they shall not silently stop/start/reset the algorithm or create a replacement algorithm session. For masked-fallback recovery, same physical pod continuity is established by persisting the fallback pod identity with the connected delivery baseline and comparing it with the recovery snapshot; different/new pod, unknown identity, or delivery-counter/history discontinuity suppresses replay while preserving the existing session/cadence state if the loop remained armed. If the previous pod is known inactive/retired or past hard service-stop before restore evidence can be collected, the recovery path preserves the existing algorithm session and records assumed modeled fallback delivery instead of creating a new session or holding the user in an unrecoverable blocked state.
- SDD-POL-029 (issued-dose attribution replay): `LoopRuntimeCoordinator` persists `PendingIssuedDoseAttribution` for any applied or uncertain automatic bolus command, not only meal announcements. The attribution records request step, first eligible attribution step, requested units, request/completion timing, component split (basal, correction, meal, or mixed), flow ID, and pod identity when available. `PumpServiceAdapter` restores that metadata before post-relaunch status reconciliation; persisted matching reconciliation evidence defers or suppresses the adapter's pod-counter reconstruction, except under the SRS-PUMP-010 fresh-completion upgrade rule: a fresh same-pod settled status taken at/after the dose's expected physical completion with the `bolusNotDelivered` register at zero supersedes the deferral and the adapter surfaces the pump-observed completed delivery (a canceled/interrupted partial's nonzero register keeps the deferral), while the engine replaces matching non-assumed evidence recorded below the requested amount with a strictly-higher fresh same-pod post-completion pump observation before that evidence is consumed. A same-pod `.delivering` or unknown state keeps the attribution pending and skips the due step under the accepted DASH completion liveness bound. Fresh matching evidence must prove request step, requested-unit tolerance, pod identity, and delivered units in `0...requested` before confirmed consumption. When historical rows are due, coordinator replays them before the live step with `CGM=-1`, injects delivered insulin only at the first eligible attribution row, and emits no catch-up pump command; when the live due step is the first eligible attribution step, it consumes the evidence directly without synthetic replay. Definitive command rejection (v1.65): the adapter records request metadata immediately before entering

OmniBLE so crash/uncertain outcomes remain recoverable. If OmniBLE returns a definitive blocked result, the coordinator calls `DefinitivelyBlockedBolusCommandHandling` with the unique request step and units. `PumpServiceAdapter` clears only a matching cached request and any matching cached `lastDelivery`; a mismatched/newer request is retained. This prevents a rejected request from being paired with DASH `bolusNotDelivered == 0` and fabricated as completed delivery. Uncertain outcomes deliberately do not invoke this cleanup. Fresh-completion implementation clarification (v1.53; supersedes the register-only exception above): `IssuedDoseReconciliationEvidence.observationContext` is decode-compatible optional durable provenance. Cancel returns `persist .canceledDelivery`; pump progress whose resolved source is capped while active persists `.deliveryInProgressCapped`; legacy records decode with no context. Adapter and engine completion upgrades require the explicit in-progress-capped context for partial evidence. Canceled and nil-context partials continue to defer reconstruction even when a later DASH refresh reports zero not-delivered. Assumed-full evidence may allow a matching direct observation to surface, but cannot increase insulin attributed to the algorithm. When fallback recovery and a pending issued dose share a same-pod cumulative pump delta, `FallbackBasalExposureReconciler` partitions the known dose only if the total covers it, completion time has passed, identity matches, and the residual credibly matches fallback exposure. The issued-dose share is persisted as `partitioned_pump_delta`; the fallback event stores residual actual delivery separately from the raw pump total. If the split is not credible, no partition is guessed and the raw total is not represented as actual fallback delivery. Issued-dose replay precedes fallback replay. Its confirmation tuple echoes the issued ask exactly, dead-gap rows use zero-delivery echo retractions, and the first fallback row echoes the preceding replay ask. If fallback begins on the attribution step itself, the resolved dose and one bounded fallback share merge into that row with the fallback share carried only in delivered units. Once a fresh settled status exists, unresolved, mismatched, non-credible, missing-identity, retired-pod, or replacement-pod cases resolve under the accepted assumed-delivered clinical policy: requested units are consumed exactly once by replay or live input, the assumed source remains explicit in UI and telemetry, and meal-linked assumed doses schedule meal-adaptation exclusion. A replacement pod's unrelated `lastDelivery` is scrubbed before live input, and later automatic commands are not blocked solely because the original pod cannot be queried. Consumed request-step deduplication is keyed by request step rather than units; a later same-step units mismatch becomes discrepancy evidence instead of fresh insulin. Legacy persisted `activeIssuedDoseReconciliationHold` keys are ignored during decode. The writer-less hold mechanism was removed in `c42f2a6`; no current clinical disposition depends on it. When Home presents an unresolved meal-delivery progress modal for a matching meal-linked pending attribution and the original pod is known unavailable/service-stopped/different or has not produced matching evidence by expected physical completion plus two minutes, `MealDeliveryPodReplacementEscapePolicy` exposes a `Replace Pod` action. The action uses `LoopRuntimeEngine.reconciledUserAbandonedMealIssuedDoseEvidenceState(...)` to persist assumed-delivered evidence with `IssuedDoseReconciliationFailure.reason`

`== user_abandoned_unavailable_pod`, clears only the meal-progress UI fields, leaves `PendingIssuedDoseAttribution` for coordinator replay/live-step consumption, emits the meal resolved telemetry path as uncertain/reconciled, and then routes to pod setup. If legacy/partial state lacks pod identity, coordinator accepts this assumed-delivered evidence only for the user-confirmed unavailable-pod path and only when request step, requested units, delivered-unit bounds, and non-active pump status match. The policy rejects correction-only or basal-only pending attributions and hides the action once matching reconciliation evidence exists.

- **SDD-POL-008 (Alert policy):** multi-source alert normalization model drives pump and CGM alert mapping plus pump signal-loss debounce (5 minutes), no-active-pod debounce (5 minutes), repeating safety-critical notification attempts every 30 minutes while the condition remains true, dedupe, deterministic alert precedence, and auto-clear behavior. Home active-alert presentation uses a severity-then-recency stack: the highest-priority alert renders in full with a compressed peek edge and an explicit "N More" expansion revealing the remaining Home-visible alerts in place. A sponsor-approved root-cause suppression matrix (2026-07-09) hides generic alerts from the Home surface while a specific active alert already explains them: `ALERT-ALGORITHM-STEPPING-INTERRUPTED` is hidden while forced-open-loop, step-realign, awaiting-first-step, pump signal-loss/no-pod/expired/fault/suspend, or CGM unavailable/failed alerts are active, and `ALERT-LOOP-COMMAND-BLOCKED` is hidden while forced-open-loop or those pump-outage alerts are active. Suppression is presentation-only (issuance, telemetry, and clearing unchanged), never applies to safety-critical alerts, and Alert Center always shows the complete active list. High-priority non-CGM alerts additionally route to background local notifications through `AppAlertCenter` with per-alert dedupe and severity cooldown (`actionable`: 5 minutes, `safetyCritical`: 60 seconds), and resolved alerts clear matching pending/delivered OS notifications. CGM availability/failure alerts are informational in-app status only and never schedule background local notifications; the FDA-cleared Dexcom application remains the source of truth for CGM alarming. Notification-tap routing maps alert category to app context (`pump` -> Pod modal, `cgm` -> CGM modal, non-device -> Home), except suspend-in-progress and suspend-ended alerts route directly to `Settings > Exercise and Insulin Delivery`, where the resume control is located; the suspend-ended in-app alert additionally carries a direct resume action. Its persisted deadline is the initial reminder timestamp plus 15 minutes; while the same condition remains active, expiry changes the alert from actionable `Resume Insulin` to safety-critical `Insulin Still Suspended`, emits an alert-update telemetry event, and schedules one fresh safety-critical notification. Countdown removal makes the transition exactly once, and a repeated source issue cannot restart or downgrade an already-escalated condition. The `cgm` route remains defined for in-app navigation/preview support even though production CGM alerts do not schedule OS notifications. Safety-critical alerts expose explicit acknowledge action on the Home alert stack and in Alert Center (`ackState == requiresAcknowledge`); current production required-ack alerts are pump fault, pump incompatible, app-derived `ALERT-CGM-URGENT-LOW`, and `ALERT-FALLBACK-REPLAY-DISPOSED` (unconfirmed backup-insulin disposition). OS-notification clear calls are idempotent on every retract attempt (including already-cleared keys) to cover async

schedule/retract races; alert lifecycle telemetry still emits `alert.notification.cleared` only when an active alert transitions to removed. App-level active/recently-cleared alert state is persisted and restored across relaunch; pump/cgm delegate `PersistedAlertStore` hooks are implemented for issued/unretracted/retracted lifecycle lookup. On no-active-pod restore/sync cleanup, only non-critical pod-tied alerts are auto-retracted; `ALERT-PUMP-FAULT` and `ALERT-PUMP-INCOMPATIBLE` are retained until explicit lifecycle retraction or acknowledge closure. Time-sensitive alert metadata (`ALERT-PUMP-EXPIRING`, `ALERT-PUMP-EXPIRED`, and suspend-ended escalation) is refreshed by `AppAlertCenter` on a minute interval while active. Pod expiration/expired alerting is sourced from both delegate issue/retract callbacks and state-driven synthesis from pump `expiresAt` on status refresh/startup to avoid relaunch misses and misclassification. For CGM, failed/expired classification is state-driven (`lifecycleState/algorithmState`) while delegate fallback keyword mapping prioritizes unavailable-temporary states and does not escalate failed/expired from message-only payload text; the only app-derived CGM required-ack path is the urgent-low review alert defined in `SDD-POL-026`. Runtime-derived `ALERT-ALGORITHM-STEPPING-INTERRUPTED` is step-based rather than CGM-receipt-based, carries blocker detail, routes notification-tap navigation to Home, and remains distinct from pump-native alerts while CGM state remains visible as informational context. Runtime-derived `ALERT-ALGORITHM-FORCED-OPEN-LOOP` presents informational-first and escalates to safety-critical advisory wording after 3 consecutive blocked steps (fingerstick-BG remedy instruction with sensor-check/study-team guidance; no reset demand), routes notification-tap navigation to Home, persists across skipped/no-telemetry publications, and clears when a later algorithm output reports forced-open inactive. The frozen alert behavior is covered by executed automated evidence; no separate formal physical alert-validation claim is retained for this submission. Mapping baseline and source inventory are maintained in Alert Inventory and Mapping.

5. Data Persistence

- `SDD-DATA-001`: runtime cadence state is persisted in `UserDefaultsLoopRuntimeStateStore`.
- `SDD-OUTAGE-001`: fingerstick-supported sustained dosing during CGM outage (clinically approved 2026-07-14). A persisted `OutageBGRunContext` (in `LoopRuntimeState`) anchors the fingerstick-due schedule to the last qualifying glucose (`OutageBGRunPolicy`: 120-minute interval, 60 when the measured value is $>200 / <80$ mg/dL); the engine re-anchors it from each executed step's accepted input, quantifies each release window at mask re-arm by updating the engaged fallback event in place (same `fallback_event_id`; duration + `PumpBasalSchedule.deliveredUnits` modeled exposure over the programmed fallback schedule, upgraded to a pump-measured confirmed total when the coordinator accumulated per-step credits, else status `resumed_without_reconciliation`; RA-020 accounting implemented per `SRS-OUTAGE-006 v1.42`), gates the outage surfaces on the `SRS-OUTAGE-002 v1.37` definition (blind executed step AND algorithm offline mode `openLoopMode == 2` / fingerstick-fed step / mid-outage persistence via the context's last qualifying source - an isolated relaunch/wake

blind step with the dropout clock intact shows nothing), and drives three alert tiers through `AppAlertCenter.updateOutageBGDueMonitoring` (pre-scheduled OS notifications for due-soon/overdue, mirror of the stepping-interruption pattern) plus a Home countdown line. The backup-basal bridge lives in the pre-command maintenance path (`performOutageBridgeTransitionIfNeeded`): on a connected forced-open step with the mask armed it actively cancels the 0 U/hr temp basal (`PumpService.cancelTempBasal`) and marks `MaskedFallbackMode.releasedForOutage` (maintenance policy suppresses renewals in that mode); on the next non-forced-open step it re-arms the mask BEFORE the step's pump command; cancel failure keeps the mask and retries, re-arm failure blocks that step's insulin command. Bridged exposure enters the algorithm's accounting live while released: `OutageDeliveryCreditSupport` feeds one synthetic per-step tuple (echo of the persisted previous ask + identity-guarded pod-counter delta) on every executed released step, and a pod-away gap (skipped steps, then a matching-identity reconnect status with pod totals) is reconciled by the standard bounded schedule-weighted fallback replay synthesized in the coordinator (`prepareReleasedOutageGapReplayIfNeeded` -> `performPendingFallbackReplayIfNeeded`; confirmed-stamped plan, SRS-RUN-006 echo chain - sponsor direction 2026-07-18, never a single reconnect-step chunk). Per-row shares are whole multiples of the algorithm's 0.01 U insulin-ledger resolution allocated by schedule-weighted largest remainder (`FallbackBasalReplayPlanner.quantizedRowUnits`; zero-share rows keep their echo/retraction role) so the C++'s booked row total equals the measured delta exactly, with schedule weights evaluated in the engaged event's recorded profile timezone (device zone fallback, mirroring the classic replay call site). The plan itself carries the credit bundle (`FallbackBasalReplayPlan.ReleasedOutageGapCredit`: measured delta, re-anchor counter and pod identity, first-row echo ask); whichever pass consumes the plan applies the baseline re-anchor, the exactly-once accumulation, and the final-row ask handoff in the same save that clears it, so a force-quit at any mid-pass persistence point replays on relaunch with the full bundle instead of double-counting; a cadence-drift skip on the synthesis pass discards the just-synthesized plan untouched and the next pass re-synthesizes against the realigned counter. No maximum-outage stop exists by design.

- SDD-DATA-002: primary and safety/fallback algorithm state blobs are persisted in separate `UserDefaultsAlgoStateStore` instances with distinct storage keys, so neither track's persisted state can be loaded into the other.
- SDD-DATA-003: step telemetry is persisted in `LoopTelemetryStore`; shared runtime state persists the latest structured fallback-basal event and the maintenance state required across relaunch (`originalProgrammedBasalSchedule`, fallback schedule/rate, safety target, pod identity, pump-total baseline/timestamp, mask-renewal and schedule-refresh timestamps/outcomes, and recovery detail). Per-step records persist Algo2015 delivered-insulin echo fields beside algorithm snapshots so reconciled accounting is auditable. `maskedFallbackRecoveryState` and `pendingFallbackBasalReplayPlan` survive relaunch and support connected retry, bounded no-command replay, or explicit unreconciled disposition. Runtime state also persists pending issued-dose attribution, credible partition evidence, the most recent consumed issued-dose delivery, delivery discrepancy, and

reconciliation failure; lifecycle expiry prevents stale records from coloring a fresh episode. Different/new-pod cases consume assumed-delivered evidence without creating a blocking hold. Primary and secondary/safety q5 nominal-basal profiles persist independently for traceable schedule generation. Resolved fallback events retain modeled exposure, actual fallback delivery only when evidence supports it, raw pump-total delta separately, pod continuity, and replay metadata. Confirmed/corrected replay rows use $CGM=-1$, schedule-weighted delivered input, and no catch-up commands; modeled exposure remains review-only when evidence cannot support allocation. Retired/expired/no-active pod fallback recovery is persisted as a separate assumed-delivered disposition: modeled expected delivery is copied into actual delivered units only with `assumed_delivered_per_clinical_policy`, and replay/review telemetry must preserve that source so assumed fallback accounting cannot be mistaken for pump-counter-confirmed delivery.

- `pendingFallbackBasalReplayPlan.firstReplayStep / lastReplayStep` denote the fallback-active replay interval, not the full disconnected gap. The stored plan includes explicit 0 U deliveries only for replay-eligible fallback-active slots.
- SDD-DATA-004: device managers are persisted via app delegates for reconnect behavior.
- SDD-DATA-005: alert lifecycle state is persisted in both `AppAlertCenter` (active + recently-cleared timeline, capped at 100 entries) and delegate-level `LoopKit PersistedAlertStore` implementations (issued/retracted records for pump/cgm sources).
- SDD-DATA-006: clinical settings are persisted in a versioned model with deterministic defaults and migration behavior; weight is normalized lbs (UI) -> kg (runtime input). The persisted clinical config now also carries the participant target-access profile, with migration default/inference rules so legacy saved target values load into either `Pregnancy` or `Standard` deterministically.

6. Requirement Allocation

SRS ID	Design Element(s)
SRS-RUN-001..007	SDD-APP-001, SDD-CORE-001, SDD-POL-001, SDD-POL-021, SDD-PUMP-001
SRS-ALG-001..009	SDD-ALG-001, SDD-QA-001
SRS-OUTAGE-001..006	SDD-OUTAGE-001, SDD-PUMP-001, SDD-ALERT-001
SRS-PUMP-011	SDD-PUMP-001
SRS-CLIN-013..015	SDD-CLIN-001

SRS ID	Design Element(s)
SRS-CLIN-016..017	SDD-CLIN-001, SDD-CLIN-002, SDD-LOG-001
SRS-CGM-001..005	SDD-CORE-001, SDD-POL-002, SDD-POL-020, SDD-CGM-001, SDD-APP-003, SDD-SIM-001
SRS-CGM-006	SDD-CGM-002, SDD-ALERT-001, SDD-LOG-001
SRS-BG-001..013	SDD-APP-001, SDD-CORE-001, SDD-BG-001, SDD-POL-010, SDD-POL-011, SDD-POL-012, SDD-LOG-001
SRS-CLIN-001..012	SDD-CLIN-001, SDD-POL-013, SDD-POL-017, SDD-POL-025, SDD-DATA-006
SRS-PUMP-001..010	SDD-PUMP-001, SDD-POL-003, SDD-POL-029, SDD-SIM-001, SDD-DATA-003
SRS-MEAL-001..006	SDD-APP-001, SDD-CORE-001, SDD-POL-004
SRS-MEAL-007..011	SDD-APP-001, SDD-PUMP-001, SDD-LOG-001, SDD-POL-023, SDD-POL-024, SDD-POL-027
SRS-MEAL-012	SDD-APP-001, SDD-POL-029
SRS-STATE-001..005	SDD-DATA-001..005, SDD-POL-005, SDD-POL-028, SDD-POL-029, SDD-PUMP-001
SRS-LOG-001..015	SDD-LOG-001, SDD-SIM-001, SDD-POL-017, SDD-POL-018, SDD-POL-024, SDD-POL-025, SDD-APP-007, SDD-POL-028, SDD-CLIN-002
SRS-UI-001..008	SDD-APP-001, SDD-POL-006, SDD-POL-007, SDD-POL-009, SDD-POL-014, SDD-POL-018, SDD-POL-019, SDD-CGM-001
SRS-UI-009	SDD-APP-008
SRS-VAL-001	SDD-CLIN-001, SDD-DATA-006
SRS-ALERT-001..019	SDD-ALERT-001, SDD-POL-008, SDD-POL-020, SDD-POL-022, SDD-POL-026, SDD-DATA-005
SRS-ALERT-020	SDD-ALERT-001, SDD-CGM-002
SRS-SEC-001..002	SDD-LOG-001, SDD-CFG-001, and the Cybersecurity Plan (current software scope)
SRS-SEC-003..009	SDD-AUTH-001, SDD-AUTH-002, SDD-POL-015, SDD-POL-016, and the Cybersecurity Plan (documented implementation; deferred from current software package)

7. Design Constraints

- iOS background execution is opportunistic; runtime requires deterministic, bounded skip and degraded behavior.
- External device SDK behavior (`OmniBLE`, `G7SensorKit`) constrains event cadence and status timing.
- Closed-loop safety policy requires avoiding manual bolus path exposure.
- The designated study build and any later controlled build must use a tracked app-workspace SwiftPM resolution and exact CryptoSwift requirement; dependency-control checks fail when the source URL, exact version, revision, or Git control drifts.

Security/auth scope note:

- SDD-AUTH-001, SDD-AUTH-002, SDD-POL-015, and SDD-POL-016 document current implementation boundaries because the code exists in the repository.
- For the current software package, those auth/provider/session-continuity design elements are documented implementation context only.
- Closure claims for SRS-SEC-003...009 remain deferred and are not being asserted through the current software package.

8. BG Design and Deferred Considerations

8.1 Functional flow

1. User opens BG entry flow and submits BG with explicit confirmation.
2. Runtime validates session state and computes `expectedStep`.
 - Submitted BG must be within 20...600 mg/dL.
3. Runtime creates/updates one pending BG candidate:
 - if due step has not executed yet, target that due step;
 - if due step already executed, target immediate next step.
 - if a candidate already exists, replace with newer BG submission.
4. Runtime dispatches `doWork(cause: bgCheck)` without borrowing future slots.
5. On execution, coordinator consumes pending BG candidate only when step index matches candidate target and candidate freshness is valid.
6. Coordinator builds algorithm input with:
 - `BGval` from submitted manual BG.
 - CGM input from normal CGM mapping policy (may remain -1).
7. Algorithm executes once for due step.
8. Command application follows normal pump safety policy.
9. Telemetry records `manualBG` source, value, timestamps, and execution outcome.
10. If candidate target step is missed without consumption, runtime discards candidate as too old.

8.2 Guards and invariants

- No future-slot borrowing is allowed for `bgCheck`.
- Exactly one pending BG candidate is kept at a time.
- New BG submit replaces existing pending candidate before consumption.
- Pending BG candidate is single-step scoped and expires if not consumed on immediate next target step.
- Manual BG freshness is required at execution time.
- Pump-unavailable behavior remains unchanged from degraded-mode policy.
- Before first successful anchored step exists, manual BG submit is rejected (step-0 BG rescue disabled by current policy).

8.3 Deferred considerations

- The current baseline uses immediate-next-step validity rather than a separate wall-clock manual-BG freshness window. A future wall-clock limit would require an approved SRS, risk, design, and verification change.
- Step-0 BG-rescue policy (SRS-BG-008) is deferred; current implementation remains CGM-only at step 0.

9. Simulation Strategy (Medium now, High later)

9.1 Medium-fidelity harness (in-scope)

- Architecture:
 - Scenario runner emits deterministic event timelines into runtime ports.
 - Mock services conform to production protocols (`CGMService`, `PumpService`) so runtime/coordinator logic remains unmodified.
 - Virtual clock controls step-slot progression (300s cadence, early window, skip/catch-up timing).
- Scenario event classes:
 - CGM readings (value, timestamp, reliability, availability transitions)
 - pump status transitions (`idle`, `delivering`, `unknown`, `disconnect/reconnect`)
 - command-result events (success/failure/reconciliation data)
 - normalized alert issue/retract events
- Required outputs:
 - executed step index and skip reason
 - algorithm input/output snapshot
 - command-application decision
 - alert center active/recent lifecycle state

9.2 High-fidelity emulator (future)

- Target:
 - BLE/session-level emulation for Omni/G7 transport behavior (timing jitter, ACK patterns, session-restore edge cases).
- Positioning:
 - not a near-term gating dependency; added after medium-fidelity harness is stable.

9.3 Safety/testing contribution

- Improves hazard-path reproducibility for race conditions that are hard to force on demand with real devices.
- Expands deterministic pre-merge coverage for gating failures (step timing, degraded inputs, command-block behavior, alert escalation).
- Produces traceable verification artifacts that can be mapped to RA/SRS/TV with lower variance than ad hoc manual runs.
- Complements, but does not replace, hardware-in-the-loop verification for true transport and physical-delivery behavior.

Controlled Software Risk Analysis

Status: Frozen-baseline risk analysis approved for IDE submission Version: 1.53 Owner:

BionicLoop engineering

Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, residual-risk acceptance and final software-package approval, controlled email dated 2026-09-07 EDT

Baseline freeze SHA: 91c0e98a9bc9429a0486bebdebfffc7d8dbbe300e Scope: BionicLoop investigational closed-loop app (Dexcom G7 + OmniPod DASH + Algo2015) Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-25	Engineering	Initial controlled draft baseline
0.9	2026-04-06	BionicLoop engineering	Added controlled-document metadata and refined software-only cybersecurity boundary language for RA-009.
0.91	2026-04-07	BionicLoop engineering	Tightened residual-risk wording for local telemetry exposure, investigational clinical gating, and reconnect-fallback behavior to match the current baseline and remaining freeze-time evidence posture
0.92	2026-04-08	BionicLoop engineering	Added masked-fallback maintenance and restore guardrail wording for the masked offline-fallback design
0.93	2026-04-08	BionicLoop engineering	Clarified masked-fallback deferred-maintenance and recoverable-restore controls for the implemented design
0.94	2026-04-14	BionicLoop engineering	Added reconciliation-required blocked-state controls after offline mask expiry for the masked-fallback design
0.95	2026-04-14	BionicLoop engineering	Added reconnect restore/disarm retry and explicit fresh re-arm controls for pending masked-fallback recovery
0.96	2026-04-14	BionicLoop engineering	Updated masked-fallback recovery risk controls to describe same-session unreconciled resume as the current feasibility path after successful restore
0.97	2026-04-15	BionicLoop engineering	Added pod total-delivery baseline and modeled-vs-pump-reported fallback review controls for the masked-fallback design
0.98	2026-04-15	BionicLoop engineering	Added confirmed/corrected basal-only reconnect replay controls, persisted replay-plan state, and replay-detail review visibility for the masked-fallback design
0.99	2026-06-02	BionicLoop engineering	Updated masked-fallback controls to pump-delta missed-step algorithm replay and added repeated no-active-pod critical alert

Version	Date	Author	Summary of Changes
			controls
1.00	2026-06-06	BionicLoop engineering	Added same-pod continuity as a replay authorization control and formalized explicit-operator-only algorithm session lifecycle
1.01	2026-06-08	BionicLoop engineering	Added schedule-weighted replay allocation controls and zero-weight no-replay behavior for credible same-pod pump delta
1.02	2026-06-11	BionicLoop engineering	Clarified that fallback replay is bounded to confirmed fallback-active missed slots and excludes pre-activation disconnected gaps
1.03	2026-06-12	BionicLoop engineering	Added generalized issued-dose attribution controls for interrupted automatic bolus commands and different/new-pod no-replay disposition
1.04	2026-06-13	BionicLoop engineering	Hardened issued-dose risk controls for non-credible delivered-unit evidence, old-pod delivery input scrubbing, and automatic resume blocking
1.05	2026-06-15	BionicLoop engineering	Updated different/new-pod issued-dose control to assume delivered without blocking replacement-pod dosing; documented possible 4-hour adaptation-forget control pending team signoff
1.06	2026-06-17	BionicLoop engineering	Added explicit meal-progress pod-replacement escape that records assumed-delivered evidence before clearing trapped UI state
1.061	2026-06-18	BionicLoop engineering	Added retired/expired/no-active-pod fallback recovery control that uses assumed modeled fallback exposure when connected pump evidence is no longer recoverable. (Renumbered 2026-07-09: this row previously duplicated version 1.06.)
1.07	2026-07-08	BionicLoop engineering	Added verification-backed liveness bound for the accepted issued-dose skip-and-wait/no-boundary-cancel policy.
1.08	2026-07-10	BionicLoop engineering	RA-014 control hardened after a field-observed dual-instance isolation defect: the safety-track instance shared the C++ process-global state, halving buffer-time safety windows (observed 34-min forced open-loop vs designed 60); resolved with symbol-isolated per-instance algorithm copies and isolation regression tests. Connected-outage zero-insulin window recorded as ANOM-003 pending clinical policy.
1.09	2026-07-10	BionicLoop engineering	Independent review: added the qualitative risk-scoring framework (severity/probability scales, risk matrix, acceptability criteria) and per-row SxP ratings; added RA-018 (connected-outage zero-insulin window, ANOM-003) and the overall residual-risk/benefit-risk conclusion; controls restated as shipped behavior (feasibility-work phrasing retired); RA-011 forced-open control aligned to the informational-first/fingerstick copy; RA-015 pending freeze decision moved to the acceptance

Version	Date	Author	Summary of Changes
			notes; RA-005 residual resolved to Low; RA-014 linked to SRS-ALG-008. Header version corrected (1.07 -> 1.09; the 1.08 row landed without a header bump).
1.10	2026-07-13	BionicLoop engineering	RA-018 acceptance note updated with the received clinical policy answer (2-hour fingerstick cadence with conditional hourly escalation, no maximum-offline hard stop); residual stays provisional pending sponsor-approved design, implementation, and verification.
1.11	2026-07-14	BionicLoop engineering	RA-018: dropout-clock characterization complete (fingerstick restarts the full 60-minute window - real-C++ pins) and the backup-basal bridge design clinically approved (Camille Powe, MD, 2026-07-14); residual stays provisional pending implementation and verification.
1.12	2026-07-14	BionicLoop engineering	RA-018 controls updated to the implemented fingerstick-supported CGM-outage dosing behavior (schedule, tiered alerts, countdown, backup-basal bridge, no-stop rationale) with residual re-stated (implemented + unit-verified; formal and physical evidence pending); RA-017 verification links gain TV-PUMP-010 (pump-command watchdog implemented).
1.13	2026-07-14	BionicLoop engineering	RA-013 controls extended per clinical IFU-review feedback: first-launch Standard/120/75%/40 defaults (never rewriting persisted values) and the New Participant Reset participant-data wipe for reused phones (also supports the RA-009 local-data posture between participants).
1.14	2026-07-14	BionicLoop engineering	Independent review alignment: RA-017 controls corrected to the implemented assumed-delivered-per-clinical-policy disposition (stale no-guess/hold wording removed; the future 4-hour adaptation mitigation moved from controls to the acceptance notes per Method) and gains SRS-PUMP-011 + the pump-command watchdog control; RA-013 armed-case wording aligned to the guided stop-then-reset flow with SRS-CLIN-013/014 and TV-CLIN-014/015 links; RA-018 gains SRS-OUTAGE-001..006 + TV-OUTAGE-001..003 links and the OS-notification-permission control gap (in-app surfacing identified as a gap); RA-014 residual rationale made explicit (harm-vs-occurrence basis after the dual-instance isolation incident, provisional); Residual Probability Rationale section added; severity-scale confidentiality note added; RA-019 added (unintended New Participant Reset); Overall Residual Risk Conclusion rewritten to enumerate the provisional rows and retire the already-made ANOM-003 clinical-disposition condition.
1.15	2026-07-16	BionicLoop engineering	RA-020 added (discovered in live field data during a sponsor outage drill): backup basal delivered during a connected CGM-outage window is not counted by the algorithm - error direction

Version	Date	Author	Summary of Changes
			is toward extra insulin, bounded by fallback rate × outage duration. Interim control shipped same day (release windows quantified on the engaged fallback event at re-arm: duration + modeled schedule exposure, resumed_without_reconciliation); planned control is algorithm-side accounting (differential replay and simulated-outage verification), blocking for any human-use build. This field test carried no exposure risk (saline, no participants).
1.16	2026-07-17	BionicLoop engineering	RA-021 added (coverage review, 2026-07-17): subject weight was the one dose-scaling clinical input with no plausibility bounds before the algorithm. Mitigated same day: SRS-CLIN-015 save-gate range 50-500 lbs, store top-clamp, and 20-230 kg algorithm last rail (TV-CLIN-016). Range flagged for sponsor confirmation.
1.17	2026-07-17	BionicLoop engineering	RA-009 re-scored High to Low (sponsor-approved closure package): file sharing removed, CSV sealed to the gated share with complete protection + backup exclusion, spool protected, redundancy role replaced by SRS-LOG-012 provable delivery with BionicScout completeness visibility. The freeze export-posture condition is resolved: decided, not deferred.
1.18	2026-07-17	BionicLoop engineering	Coverage review made the RA-020 mechanism statement precise by naming phantom assumed-delivered mode-2 asks and uncounted backup basal, with drill-1 net-direction evidence and recorded-sequence differential quantification. Extended RA-019 controls for the SRS-CLIN-014 v1.41 telemetry drain guard with a pre-confirmation final flush, undelivered-count consent line, best-effort <code>telemetry.outbox.discarded</code> marker before erasure, and sequence-continuity-preserving erasure.
1.19	2026-07-18	BionicLoop engineering	RA-020 controls extended (sponsor-directed, 2026-07-18 field incident): the accounting control is implemented, and pod-away gap reconciliation within a released window is recorded as replay-based (time-positioned schedule-weighted rows via the standard bounded fallback replay; baseline re-anchor; exactly-once accumulation) rather than a single reconnect-step chunk - the chunk positions the whole gap delta at the pre-gap request step, and the measured real-C++ divergence (<code>Algo2015OutageGapPodAwayReplayCharacterizationTests</code>) is cited in the controls cell.
1.20	2026-07-18	BionicLoop engineering	RA-020 controls corrected from the independent safety review of the gap-replay work: row shares restated as 0.01 U-quantized largest-remainder allocation with engine-ledger conservation pinned against the real C++ on the field and overnight shapes (fractional shares had inflated the C++ ledger +0.03 U / +0.14 U; sponsor sign-off flagged for the quantized-time-positioning

Version	Date	Author	Summary of Changes
			policy)); the credit context now persists inside the replay plan, closing the force-quit-at-mid-pass-save double-count (reviewer probe fed 0.29999 U for a physical 0.15 U; now conserved exactly); measured chunk-vs-replay numbers refreshed for the quantized lane (0.1235 vs 0.0811 U reconnect IOB, two 0.05 U batches freed earlier, residual gap bounded by dosed divergence).
1.21	2026-07-18	BionicLoop engineering	Documentation truth-sync: RA-020 is restated as an implemented control rather than a planned control; residual closure remains conditional on sponsor sign-off for the 0.01 U quantized time-positioning policy, the exact hardware re-drill, independent safety-review completion, merge designation, and formal-lane execution. RA-009 acceptance and the overall residual-risk conclusion are aligned to the sponsor-approved 2026-07-17 closed posture, including implemented protection/backup exclusion for enumerated Algo2015 diagnostics.
1.22	2026-07-19	BionicLoop engineering	RA-004/011/012 controls aligned to exact-step pending-fingerstick meal execution and same-severity pump-signal precedence: combined BG + meal is real-C++-characterized and dual-track pinned; stale/invalid/pump-unavailable negative paths remain blocked; signal loss cannot be masked by a newer check-BG prompt.
1.23	2026-07-20	BionicLoop engineering	RA-010/011/012/018 controls aligned to the spatially separate exact-value Manual BG review and the shared Home gold emphasis for direct check-BG, due-soon, and overdue fingerstick requests.
1.24	2026-07-21	BionicLoop engineering	RA-017 issued-dose completion evidence hardened after review against the 2026-07-08 cancel/relaunch incident: a zero DASH bolusNotDelivered register cannot overwrite persisted partial-delivery evidence; only explicitly persisted in-progress-capped provenance can authorize a higher settled observation. Canceled and legacy-unclassified partials remain authoritative, with regression coverage at the adapter, runtime, and persistence boundaries.
1.25	2026-07-21	BionicLoop engineering	RA-004 control closes the borrowed-step forced-open race: the last qualifying glucose step is persisted, prospective meal availability blocks the characterized 13th blind step, and the coordinator repeats the check before accepted-meal persistence. Fresh CGM and valid exact-step BG remain the only glucose exceptions; rejection creates no meal/algorithm/pump action and gives explicit no-delivery feedback.
1.26	2026-07-22	BionicLoop engineering	RA-004 participant guidance corrected for connected busy pumps: active <code>.delivering</code> status now produces wait-for-delivery guidance even when attribution is pending; reconnect

Version	Date	Author	Summary of Changes
			guidance remains for idle unresolved delivery. The blocking control and dosing behavior are unchanged.
1.27	2026-07-24	BionicLoop engineering	Added RA-022 for an incorrect, absent, stale, or safety-profile-mixed Temporary Target, with persisted absolute expiry, execution-time primary resolution, permanent-target safety/fallback separation, explicit activation gates, UI/notification visibility, reset handling, and app-to-Scout telemetry controls.
1.28	2026-07-30	BionicLoop engineering	Extended RA-010/RA-011 controls for relocating the existing suspend/resume action to Exercise and Insulin Delivery, removing the duplicate Pod Settings action, preserving manual-resume reminders and disabled unsafe states, and routing suspend-alert actions to the control.
1.29	2026-07-30	BionicLoop engineering	Extended RA-010 controls so Manual BG and Meal Announcement remain persistently visible in the Home bottom safe area while variable-length status, alert, and chart content scrolls.
1.30	2026-07-30	BionicLoop engineering	Extended RA-011 controls so an ended suspension reminder provides a direct serialized resume action and escalates to a fresh safety-critical alert/notification after 15 minutes if insulin remains suspended.
1.31	2026-07-30	BionicLoop engineering	Extended RA-004/RA-017 controls for suspended-meal rejection before algorithm execution and matching-only cleanup of definitively blocked bolus metadata so DASH status reconstruction cannot fabricate and repeatedly feed an undelivered meal dose.
1.32	2026-07-30	BionicLoop engineering	Added RA-023 for stale old-sensor binding and wrong-nearby-sensor adoption, controlled by explicit persistent replacement mode, staleness non-adoption, authenticated discovery, participant guidance, durably deduplicated telemetry, and required multi-device physical validation.
1.33	2026-07-31	BionicLoop engineering	Extended RA-015 controls so interruption guidance cannot retain a stale CGM diagnosis when suspension or fallback recovery actually blocked stepping; blocker state now follows the latest unresolved outcome and clears after successful live execution. Extended RA-019 reset controls so asynchronous G7 persistence/telemetry completion cannot recreate outgoing-participant state.
1.34	2026-08-06	BionicLoop engineering	Extended RA-008 auditability controls so Recent Dose Steps identifies valid fingerstick input, displays its exact value, and preserves concurrent CGM evidence.
1.35	2026-08-11	BionicLoop engineering	Extended RA-014 controls after field telemetry identified a host mapping defect that omitted concurrent basal insulin from meal-

Version	Date	Author	Summary of Changes
			step commands and caused explicit Algo2015 reconciliation rejection.
1.36	2026-08-11	BionicLoop engineering	Aligned RA-013 with the implemented clinical-configuration boundary (staff initial setup only while configuration is unset; offline unlock for all later access/edits) and removed obsolete development labels from the implemented RA-020 control.
1.37	2026-08-13	BionicLoop engineering	Extended RA-009 controls after field telemetry backlog amplification: update-safe incremental spool migration, retained-evidence-first recovery, aggregate chatter-drop reporting, transition-only notification telemetry, and passive backlog visibility; no score change.
1.38	2026-08-17	BionicLoop engineering	Strengthened RA-009 controls with an uncapped retained-clinical SQLite ledger, durable-enqueue/network separation, retry liveness, bounded batch recovery with single-route fallback, and complete protected active-session step evidence; no score change.
1.39	2026-08-20	BionicLoop engineering	Extended RA-012 controls so pending fingerstick BG is visibly distinct from completed algorithm use, with persisted submission time, evidence-only fill, and same-target deduplication; no score change.
1.40	2026-08-20	BionicLoop engineering	Added RA-024 for wrong-session or internally inconsistent legacy Algo2015 recovery export; extended RA-009 local-data controls to the clinical-gated protected ZIP staging/share path.
1.41	2026-08-20	BionicLoop engineering	Revised RA-009/024 controls for the combined CSV/Algo2015 ZIP shared through Recent Dose Steps; exact missing epochs now produce CSV-only output and competing Settings presentations were removed.
1.42	2026-08-20	BionicLoop engineering	Extended RA-015 controls so a proven fresh idle Pod refresh replaces stale reconnect guidance without waking the algorithm, while stale/non-idle/no-Pod evidence cannot falsely claim recovery; no score change.
1.43	2026-08-20	BionicLoop engineering	Extended RA-010 controls with point-only CGM chart rendering so discrete sensor samples are not visually interpolated by a connector line or area fill; no score change.
1.44	2026-08-21	BionicLoop engineering	Bound the controlled risk analysis to the immutable 2026-08-21 software freeze; clarified that engineering execution is complete while sponsor approval and residual-risk acceptance remain pending. No hazard, control, score, or product behavior changed.
1.45	2026-08-21	BionicLoop engineering	Synchronized current row conclusions and overall benefit-risk status to the exact-freeze execution report. STP-ALG-001 passed behaviorally with ALG-DEV-SA-006; app/SIM/scoped-

Version	Date	Author	Summary of Changes
			security execution completed with the identified UI and cybersecurity deviations still open. Superseded candidate states remain in this revision history. No hazard, control, score, or product behavior changed.
1.46	2026-08-25	BionicLoop engineering	Recorded Camille Powe, MD's 2026-08-21 written clinical confirmation of 0.01 U schedule-weighted allocation with total conservation, RES-004 fallback activation when CGM freshness cannot be established, and the 50...500 lb participant weight-entry range with the 20...230 kg algorithm rail. Clinical policy confirmation is closed; organizational residual-risk acceptance and retained physical-claim decisions remain sponsor decisions. No hazard, control, score, or product behavior changed.
1.47	2026-08-25	BionicLoop engineering	Added the exact-freeze 6/6 algorithm linkage companion and corrected 44/44 serial UI companion, closing ALG-DEV-SA-006, AUTO-DEV-001, and AUTO-DEV-002. Original execution records remain unchanged. No hazard, control, score, or product behavior changed.
1.48	2026-08-27	BionicLoop engineering	Recorded clinical approval of the first-step Dexcom requirement, non-synthesized CGM-gap handling, investigational Clinical Settings access, supportive Scout role and official study sources, physical-use and human-factors claim boundaries, and protocol correction for the distinct fingerstick mode. No hazard, control, residual score, product behavior, or formal result changed.
1.49	2026-08-27	BionicLoop engineering	Replaced internal development labels, test-device shorthand, obsolete package language, and an unselected future-mitigation note with reviewer-neutral current-state descriptions. No hazard, control, score, or product behavior changed.
1.50	2026-08-28	BionicLoop engineering	Corrected the residual risk-level labels for RA-020, RA-021, and RA-022 from Low to Medium so they match the controlled S3×P1 matrix cell. Severity, probability, controls, verification, and product behavior are unchanged; residual-risk acceptance remains an approval action.
1.51	2026-09-03	BionicLoop engineering	Proposed deferring the RA-023 controlled multi-device G7 bench exercise from submission assembly until before first participant deployment, subject to dated Sponsor/Clinical approval and a no-participant-deployment condition until testing and final disposition are complete. Clarified that the app has no authoritative Dexcom handoff identity and that authenticated acquisition alone does not identify the intended participant sensor. No hazard, control, severity, probability, residual score, product behavior, or verification result changed.
1.52	2026-09-03	BionicLoop engineering	Recorded the Build 843 RA-023 adversarial review: CGM-screen reentry can replace the persisted acquisition manager;

Version	Date	Author	Summary of Changes
			concurrent candidates lack an atomic single-winner adoption gate; post-connection comparison is detection rather than a gate on first-reading use; and device-lifecycle replacement may begin without planned physical isolation. Expanded D06 pre-deployment disposition and test conditions. No product source, formal test result, or existing risk score changed; final residual-risk acceptance remains open.
1.53	2026-09-03	Software Developer	Added the required RA-023 probability discriminator and sponsor sub-decision, removed deferred BG controls from the active RA-012 claim, and aligned the residual rationale with the supportive-Scout boundary. The provisional score is unchanged pending D06.

Method

This risk analysis follows a qualitative process consistent with ISO 14971 concepts, scoped to the investigational software baseline. Each hazard is tracked with:

- RA-ID
- hazard statement
- initial (pre-control) risk level with severity × probability rating
- controls (implemented behavior only; pending or planned work is never listed as a control)
- linked requirements (SRS-*)
- linked verification (TV-* / manual protocol IDs)
- residual (post-control) risk level with severity × probability rating

Severity scale

Rating	Name	Definition (harm to the study participant)
S3	Critical	Could cause severe hypoglycemia, or hyperglycemia/DKA requiring medical intervention (incorrect, missing, or unaccounted insulin delivery; missed safety-critical alert)
S2	Serious	Could cause reversible or temporary harm, delayed therapy, or loss of the data/oversight needed to protect the participant
S1	Minor	Nuisance or inconvenience with no plausible path to participant harm

Confidentiality/data-exposure harms (RA-009, RA-013, RA-019) are rated on this same participant-harm scale through their downstream effect (S2: loss of the data or oversight needed to protect the participant); no separate confidentiality severity dimension is maintained at this investigational scale.

Probability scale (qualitative, per study exposure)

Rating	Name	Definition
P3	Frequent	Expected to occur routinely during the study in the absence of controls
P2	Occasional	Plausible during the study; precipitating conditions have been observed in development or field use
P1	Improbable	Not expected during the study with controls in place; requires multiple independent failures or deliberate misuse

Risk level matrix

	P1 Improbable	P2 Occasional	P3 Frequent
S3 Critical	Medium	High	High
S2 Serious	Low	Medium	High
S1 Minor	Low	Low	Medium

Acceptability criteria

- High residual risk is not acceptable by default: it requires further risk control, or documented explicit acceptance with rationale recorded in this document and confirmed at freeze designation.
- Medium residual risk is acceptable when the listed controls are implemented and linked to passing verification.
- Low residual risk is acceptable.

Register entries are shown as Level (S×P_y). Initial ratings assume no software controls; residual ratings assume the listed controls with their linked verification. Probability rates the occurrence of harm (hazard occurrence and failure of the remaining safeguards), not merely of the triggering condition.

Risk Register

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
RA-001	Loop step not executed on expected cadence due to iOS wake variability	High (S3×P3)	CGM-triggered doWork, bounded wake-cause policy, cadence anchor, skip logic, stale-state visibility	SRS-RUN-001, SRS-RUN-002, SRS-RUN-003, SRS-UI-001	TV-RUN-001, TV-RUN-002, TV-RUN-003	Medium (S3×P1)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
RA-002	Invalid CGM values used as real glucose input	High (S3×P2)	Out-of-range sanitize to unavailable (–1), step-0 gate, degraded-step safety policy, explicit blocked-state messaging	SRS-CGM-001, SRS-CGM-002, SRS-CGM-003, SRS-CGM-004	TV-CGM-001, TV-CGM-002, TV-CGM-003, TV-CGM-004	Medium (S3×P1)
RA-003	Pump unavailable causes incorrect delivery behavior	High (S3×P2)	Degraded algorithm input with unavailable pump fields, command block on unavailable pump, auto-refresh delivery-state convergence, and closed-loop-only command exposure (no manual bolus path). The masked offline-fallback subsystem adds layered guardrails: maintenance arm/refresh is gated to active algorithm sessions with a newly completed successful step, newly computed first-arm fallback is attempted before the same step's pump command so step 0 bolus delivery cannot outrun fallback schedule programming and masking, pre-command first-arm maintenance skips unavailable/unknown refreshed pump status instead of bypassing normal pump-command classification, pre-execution arm/renew/refresh maintenance requires a fresh idle pump-status refresh and defers without pod mutation when status is unavailable/unknown/delivering or refresh fails, disruptive schedule writes are blocked during active bolus / unsafe / unknown pump-delivery states, an already-active 0 U/hr mask enters renewal in the final 20 minutes of its 30 minute window and blocked renewals are treated as deferred while the current mask is still active, clean first-arm blocks leave fallback unarmed but do not automatically suppress the normal step command or retry again in the same cycle, any remask failure after possible schedule mutation or mask replacement, including late post-schedule pump blocks and existing-mask renewal failures, persists reconciliation-required recovery and blocks normal command flow, the mask may be allowed to expire only when successful loop stepping stalls or deferred maintenance runs out of remaining mask time so fallback insulin can autonomously resume, offline mask expiry now persists reconciliation-required recovery state instead of allowing ordinary authoritative live stepping to continue, reconnect/start/foreground now retry connected restore/disarm while that recovery remains pending, confirmed/corrected basal-only reconnect evidence now drives no-command missed-step primary/secondary algorithm replay only for missed delivery slots overlapping confirmed fallback-active time using CGM=–1 and per-step pump-delivery allocations only from credible pump-reported delivered insulin with proven same-pod continuity, disconnected missed slots before offline fallback activation are left missing rather than replayed as synthetic delivery rows, replay allocation weights those pump-reported units by the persisted programmed fallback schedule rather than by a scalar guess, positive pump delta with missing/zero	SRS-PUMP-001, SRS-PUMP-002, SRS-PUMP-005, SRS-PUMP-006, SRS-PUMP-007, SRS-PUMP-008, SRS-PUMP-009, SRS-RUN-006	TV-PUMP-001, TV-PUMP-002, TV-PUMP-005, TV-PUMP-006, TV-PUMP-007, TV-PUMP-008, TV-RUN-008	Medium (S3×P1)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			schedule weight suppresses replay, invalid/not-required/stale pending replay plans are cleared after trace emission rather than retried against later wakes, ambiguous/new-pod/unknown-identity/history-discontinuous reconnect evidence remains explicitly marked as unreconciled/no-replay for operator review instead of being treated as authoritative reconciliation, and retired/expired/no-active old-pod recovery may use persisted schedule-modeled exposure as assumed delivered with explicit assumed-delivered labeling when pump-counter evidence is no longer recoverable. Broader ambiguous/non-basal reconstruction is explicitly deferred from this baseline (see the scope and deferred-items register).			
RA-004	Meal announce accepted while pump not safe/ready	High (S3×P2)	Meal announce pump-ready gating requires an active Pod with known idle delivery state; unavailable-reason messaging distinguishes delivering, suspended, signal-loss, and reconciliation states. A live <code>.delivering</code> status presents wait-for-delivery guidance; <code>.suspended</code> presents resume-before-meal guidance; idle unresolved attribution retains reconnect guidance. The app persists the algorithm step that consumed the last qualifying glucose and blocks a borrowed/due meal step that would cross the characterized 12-step blind tolerance, both in UI preflight and again in the coordinator before meal persistence. Fresh CGM or a valid pending manual BG assigned to the exact prospective meal step may permit that step so glucose and meal share the input; stale, future, invalid, pump-unavailable, signal-loss, suspension, and reconciliation states remain blocked. A suspension rejection performs no meal persistence, algorithm execution, cadence advancement, pump command, or delivered-dose chart entry; pump-confirmed resume revalidates availability without waiting for a loop step.	SRS-MEAL-001, SRS-MEAL-002, SRS-MEAL-003, SRS-MEAL-004, SRS-MEAL-005	TV-MEAL-001, TV-MEAL-002, TV-MEAL-003, TV-MEAL-004, TV-MEAL-005, TV-MEAL-006	Medium (S3×P1)
RA-005	Dose quantization mismatch with DASH minimum deliverable dose	Medium (S2×P2)	Algorithm constants + command quantization validation + telemetry reconciliation	SRS-PUMP-003, SRS-LOG-001	TV-PUMP-003, TV-LOG-001	Low (S2×P1)
RA-006	App relaunch/reset causes state mismatch and unsafe cadence	High (S3×P2)	Persisted runtime/algorithm/device-manager state, explicit full reset semantics, explicit-operator-only algorithm start/stop/reset/session replacement, and for the masked offline-fallback subsystem additional persisted maintenance state capturing original programmed basal schedule, current fallback schedule/rate, safety target, connected pod identity plus total-delivery baseline/timestamp, maintenance timestamps, restore-failed or reconciliation-required recovery context, and any pending pump-delta reconciliation state so refresh/disarm decisions, reconnect/start/foreground restore retry, post-expiry blocked-state recovery, confirmed/corrected missed-step algorithm replay, and ambiguous logging-only resume	SRS-STATE-001, SRS-STATE-002, SRS-STATE-003, SRS-STATE-004, SRS-STATE-005	TV-STATE-001, TV-STATE-002, TV-STATE-003, TV-STATE-004, TV-STATE-005	Medium (S3×P1)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			remain reconstructible across app/session boundaries without silently creating a new session			
RA-007	Home pod status stale despite reconnect	Medium (S2xP2)	Connection-driven refresh path, observer wiring	SRS-PUMP-004	TV-PUMP-004	Low (S2xP1)
RA-008	Missing auditability of algorithm inputs/outputs/com mands	High (S2xP3)	Per-step telemetry capture and traceability mapping. Recent Dose Steps derives a visible fingerstick marker and exact manual BG value from the persisted algorithm input and retains concurrent CGM evidence, avoiding ambiguity about which glucose inputs the step used. The masked offline-fallback subsystem extends operator review coverage with explicit fallback lifecycle events (armed, maintenance deferred, mask renewed, schedule checked unchanged, schedule refreshed, mask expired offline, disarmed, restore/remask failed) carrying fallback rate, source, safety target, duration, reconciliation status, pod-continuity result, failure detail, modeled fallback delivery, pump-reported delivered insulin delta when the reconnect baseline is proven same-pod and available, persisted fallback schedule metadata, schedule-weighted pump-delta reconciliation detail when confirmed/corrected reconnect recovery occurs, explicit assumed-delivered status when retired/expired/no-active pod recovery uses modeled fallback exposure because pump evidence is unrecoverable, replay rows only for reconciled or assumed fallback-active missed slots, and no-replay reason when replay cannot be safely allocated. Pre-activation disconnected gaps remain absent from replay telemetry rather than being represented as delivered-insulin rows.	SRS-LOG-001, SRS-LOG-002, SRS-LOG-009	TV-LOG-001, TV-LOG-002, TV-LOG-009	Medium (S2xP2)
RA-009	Investigational telemetry artifacts stored or exported locally are accessed outside intended site/operator controls	High (S2xP3, pre-mitigation)	Closed-posture controls (2026-07-17, sponsor-approved): OS file sharing removed (UIFileSharingEnabled/LSSupportsOpeningDocumentsInPlace absent, pinned by test); CSV export sealed to Application Support with complete file protection, backup exclusion, and legacy Documents cleanup; telemetry spool protected until-first-unlock + backup-excluded; enumerated C++-created Algo2015 diagnostics receive complete-until-first-unlock protection and backup exclusion while remaining unavailable through Files sharing; the combined recovery export snapshots the CSV and matching/current enumerated algorithm session into a complete-protected, backup-excluded temporary directory, shares only through the clinical-gated Recent Dose Steps ShareLink, and removes temporary data when that view closes; the CSV's silent-redundancy role is replaced by provable delivery (SRS-LOG-012 sequence-numbered envelopes) with per-subject completeness and stale-gap alerts on BionicScout, so loss is evident before any reset. Residual boundary: applying attributes to C++-created diagnostics is best-effort after app enumeration and requires freeze-build inspection.	SRS-SEC-001, SRS-SEC-002, SRS-LOG-012, SRS-LOG-015	TV-SEC-001, TV-LOG-012, TV-LOG-015	Low (S2xP1; export posture decided - no ungated local access in any baseline)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
RA-010	User workflow errors (profile, meal UI, status interpretation, stale/unreliable CGM value interpretation, incorrect device date/time leading to misleading recency context)	Medium (S2×P2)	Explicit UI states, spatially separate exact-value Manual BG review, persistent bottom-safe-area Manual BG and Meal Announcement actions while variable-length Home content scrolls, startup-cancel paths, background-auto-cancel for meal composer, stale/unreliable-CGM display masking (-- and no trend arrow after 11 minutes or when <code>hasReliableGlucose == false</code>), boundary CGM rendering (LOW/HIGH), stepped CGM chart scaling (300/350/400) with discrete point-only samples rather than a visually interpolated connector/area, UTC clock-drift checks with actionable warning (>10 min skew, 24-hour warning rate limit), and SOP-driven usability checks. The existing Pod suspend/resume command is directly available on the less-prominent Exercise and Insulin Delivery screen and omitted from normal Pod Settings to prevent duplicate control locations; the action is disabled without a usable Pod or during suspension-state transitions, and reminder copy states that insulin remains suspended until manual resume.	SRS-UI-001, SRS-UI-002, SRS-UI-003, SRS-UI-004, SRS-UI-005, SRS-UI-006, SRS-UI-007, SRS-UI-008, SRS-CLIN-016, SRS-VAL-001, SRS-LOG-006	TV-UI-001, TV-UI-002, TV-UI-003, TV-UI-004, TV-UI-005, TV-UI-007, TV-UI-008, TV-UI-009, TV-UI-010, TV-CLIN-017, TV-LOG-006	Medium (S2×P2)
RA-011	Critical device/runtime alerts are missed, delayed, or masked by lower-severity messages	High (S3×P2)	Multi-source alert normalization; severity-first precedence; a narrow same-severity rank that keeps Pod signal loss above a newer check-BG prompt while leaving every safety-critical alert above both; debounce/coalescing rules; repeated no-active-pod notification attempts; explicit ack/clear policy; protocol-aligned wording; suspend reminders explicitly state that insulin remains suspended until manual resume, offer the existing resume command directly on the active alert with duplicate-command suppression and retryable failure, remain until pump-confirmed recovery, route notification taps to Settings > Exercise and Insulin Delivery, and escalate with a fresh safety-critical alert/notification after 15 minutes still suspended; direct check-BG, due-soon, and overdue requests visibly emphasize the Home BG-entry control, with haptic feedback for Algo2015 checkBG; and Algo2015 forced-open-loop command-block alerting (informational-first, safety-critical escalation after 3 consecutive blocked steps, fingerstick-BG remedy instruction with sensor-check/study-team guidance; no reset demand)	SRS-ALERT-001, SRS-ALERT-002, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-ALERT-006, SRS-ALERT-011, SRS-ALERT-017, SRS-ALERT-018, SRS-ALERT-019, SRS-CLIN-016	TV-ALERT-001, TV-ALERT-002, TV-ALERT-003, TV-ALERT-004, TV-ALERT-005, TV-ALERT-009, TV-ALERT-010, TV-ALERT-016, TV-ALERT-017, TV-ALERT-018, TV-CLIN-017	Medium (S3×P1)
RA-012	Manual BG entry is stale, invalid, duplicated, misapplied to wrong step, or executed while loop is off, causing incorrect algorithm input or user confusion	High (S3×P2)	Dedicated bgCheck policy with single pending candidate, immediate-next-step-only validity, replace-on-new submit semantics, explicit stale/duplicate guards, loop-armed gating, accepted manual BG range 20...600 mg/dL, spatially separate exact-value review that cannot be bypassed by repeated taps on the entry action, source-tagged telemetry, and explicit user feedback for rejected BG actions. Home renders accepted persisted pending BG as an unfilled marker at submission time and fills it only from completed algorithm-input evidence; original-time retention and same-target deduplication prevent a consumed value from moving or appearing twice, while	SRS-BG-001, SRS-BG-002, SRS-BG-003, SRS-BG-004, SRS-BG-005, SRS-BG-006, SRS-BG-007, SRS-BG-009, SRS-BG-010, SRS-BG-011,	TV-BG-001, TV-BG-002, TV-BG-003, TV-BG-004, TV-BG-005, TV-BG-006, TV-BG-008, TV-BG-009, TV-BG-010, TV-BG-011, TV-BG-012, TV-BG-013, TV-MEAL-002	Medium (S3×P1)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			pending-state disappearance is never treated as proof of use. When a meal is accepted for that exact pending target step, the same BG and meal inputs are delivered to both primary and safety algorithms and the BG candidate is consumed once only after the shared step succeeds.	SRS-BG-012, SRS-BG-013		
RA-013	Clinical settings misconfiguration or unauthorized access leads to unsafe algorithm/session configuration (target/upfront/TMAX/subject/weight/s tart-reset controls)	High (S3xP2)	Dedicated staff initial-configuration surface available only while no usable subject configuration exists, followed by a clinician-gated settings surface; subject-scoped offline clinical unlock verifier; validated app-side material installation/refresh into Keychain-backed per-subject verifier material/local state; single-use monotonic counter burn before unlock; accepted-counter preservation across material refresh; active-unlock clearing on material refresh; local unlock expiration/manual lock with storage-failure visibility; malformed/non-ASCII/reused/wrong-subject/beyond-lookahead/unsupported-version rejection; no production master secret in app source; strict selector bounds/step validation; lbs->kg normalization; persisted defaults/migration; clinician-controlled participant target-access profile (Pregnancy / Standard); required approval capture before participant target changes; profile-bounded clinician target options with nearest-allowed normalization on profile switch; and unchanged start/reset behavior semantics under relocated controls. First-launch defaults follow clinical direction 2026-07-14 (Standard profile, 120 mg/dL, 75% meal upfront, 40-minute TMAX for Fiasp), never rewrite persisted values, and can be saved without an unlock only until the first usable configuration is committed; every later access/edit requires the offline unlock. An unlock-gated, dual-confirmed New Participant Reset wipes all participant-scoped data (including subject-scoped unlock material and telemetry) and signs out before a reused phone reaches the next participant; the reset is refused while a pod is active, and when a session is armed the control offers a guided stop-then-reset flow (explicit session stop, then the erase confirmation) rather than a silent stop or an unexplained refusal. This remains an investigational single-device clinical unlock control and is not a full production role-based authorization system.	SRS-CLIN-001, SRS-CLIN-002, SRS-CLIN-003, SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006, SRS-CLIN-007, SRS-CLIN-008, SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-011, SRS-CLIN-012, SRS-CLIN-013, SRS-CLIN-014, SRS-VAL-001, SRS-LOG-008	TV-CLIN-001, TV-CLIN-002, TV-CLIN-003, TV-CLIN-004, TV-CLIN-005, TV-CLIN-006, TV-CLIN-007, TV-CLIN-008, TV-CLIN-009, TV-CLIN-010, TV-CLIN-011, TV-CLIN-012, TV-CLIN-013, TV-CLIN-014, TV-CLIN-015, TV-LOG-008	Medium (S3xP1; accepted investigational residual; production authorization hardening deferred)

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
RA-014	Algorithm logic regression or latent C++/host-interface defects remain undetected due to insufficient structural/deterministic test coverage, leading to incorrect dosing recommendations under specific input sequences	High (S3×P2)	Dedicated Algo2015 structural-coverage campaign, deterministic golden-vector replay suite, bridge-contract edge-case testing, persistence/restart continuity testing, mandatory static-analysis lane on formal runs, and risk-based conditional MISRA closure (report+deviations or explicit N/A rationale) with IDE-traceable STR evidence packaging; symbol-isolated per-instance algorithm copies so the safety/fallback track shares no C++ global state with the primary (dual-instance isolation correction, 2026-07-10; isolation tests pin the buffer clock, the 60-minute dropout window, and CGM-history density); and a reference-host contract regression proving every pump recommendation includes the exact bolus + basal + meal output total and that matching feedback produces accepted $\bar{I}$ reconciliation rather than $\sim I$ rejection	SRS-ALG-001, SRS-ALG-002, SRS-ALG-003, SRS-ALG-004, SRS-ALG-005, SRS-ALG-006, SRS-ALG-007, SRS-ALG-008, SRS-ALG-009	TV-ALG-001, TV-ALG-002, TV-ALG-003, TV-ALG-004, TV-ALG-005, TV-ALG-006, TV-ALG-007, TV-ALG-008, TV-ALG-009, TV-ALG-010, TV-ALG-011, TV-ALG-012	Medium (S3×P1; both identified mechanisms are eliminated and regression-pinned. Exact-freeze STP-ALG-001 passed behaviorally; the 6/6 linkage companion closes ALG-DEV-SA-006.)
RA-015	The armed loop stops achieving timely successful steps and scheduled dosing recommendations may remain unserved without actionable user escalation, or inaccurate blocker guidance directs the participant toward an unrelated recovery action	High (S3×P2)	Current baseline monitors successful-step continuity rather than CGM receipt alone: while armed, the app computes a 15-minute interruption deadline from the most recent successful step (or session start / first-step wait state if none exists yet), raises ALERT-ALGORITHM-STEPPING-INTERRUPTED with local notification when that deadline is missed, preserves Home status visibility, and clears interruption state on the next successful step or loop disarm/reset. Blocker-specific guidance is derived from the latest unresolved runtime outcome or authoritative pump-native state rather than broad CGM lifecycle alerts; explicit suspension outranks fallback recovery, fallback recovery replaces stale CGM/pump-status detail, successful live execution clears persisted diagnosis, and Home suppresses the redundant interruption banner while the stronger suspend-ended alert is active without removing Alert Center/telemetry evidence. After a pump-status-unavailable attempt, only a newer idle status for an active Pod replaces reconnect guidance with communication-restored/waiting-for-glucose copy; the status refresh does not wake the algorithm or issue a command, stale/non-idle/no-Pod evidence cannot claim recovery, and a later unknown status restores unavailable guidance. Runtime also permits guarded reconnect-based fallback execution after the first anchored step exists using the current due step only without re-anchoring or backlog replay.	SRS-CGM-005, SRS-RUN-004, SRS-RUN-005, SRS-ALERT-013, SRS-ALERT-014, SRS-UI-001, SRS-UI-002	TV-CGM-005, TV-RUN-004, TV-RUN-005, TV-RUN-006, TV-RUN-007, TV-ALERT-012, TV-ALERT-013, TV-UI-001	Medium (S3×P1); exact-freeze AUTO and SIM execution passed, clinical confirmation of RES-004 is recorded, and residual risk is accepted under D07. No separate physical recovery claim is made.
RA-016	Meal announce is duplicated, silently lost, or ambiguously applied because submit/outcome state is not durable across relaunch, comms interruption, or competing triggers	High (S3×P2)	Implemented baseline now persists pending meal-request state plus correlated meal <code>flow_id</code> across relaunch, blocks duplicate meal entry while the intended execution step remains unresolved, defers Home success messaging/telemetry until runtime/coordinator result is known, blocks repeat meal entry with explicit operator guidance when command outcome is uncertain until reconciliation or session reset/disarm, rejects competing-trigger slot conflicts with explicit retry guidance instead of silently reassigning meal	SRS-MEAL-007, SRS-MEAL-008, SRS-MEAL-009, SRS-MEAL-010, SRS-MEAL-012, SRS-LOG-007, SRS-UI-002	TV-MEAL-008, TV-MEAL-009, TV-MEAL-010, TV-MEAL-011, TV-MEAL-013, TV-LOG-007	Medium (S3×P1); exact-freeze AUTO and SIM/pod-simulation execution passed. Residual risk is accepted under D07; no separate physical meal-

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			intent to a later borrowed step, provides an explicit user-confirmed pod-replacement escape for unconfirmed old-pod meal delivery after timeout or known pod unavailability, and emits accepted / resolved telemetry closure when the request is accepted, reconciles after uncertainty, or is cleared by session reset. Loop command telemetry preserves uncertain-vs-blocked semantics for cloud reconstruction.			lifecycle claim is made
RA-017	Automatic bolus delivery is interrupted or unreconciled, causing the primary/secondary algorithms to miss delivered insulin or issue additional insulin before prior delivery is safely dispositioned	High (S3×P2)	Runtime now persists issued-dose attribution for any applied or uncertain automatic bolus command, including meal, correction, mixed, and basal micro-dose commands. Active delivery blocks live step advancement. The accepted no-boundary-cancel policy is bounded by fresh DASH delivery-state convergence: supported automatic bolus sizes complete before the next 5-minute step under connected status, and disconnect/expiry/no-active-pod paths leave the active-delivery skip path by reporting unavailable/unknown status rather than <code>.delivering</code> . Matching same-request same-pod pump evidence replays missed primary/secondary steps with <code>CGM=-1</code> , actual delivered insulin injected only at the first missed attribution step, no catch-up commands, and monotonic/non-duplicating coordination with fallback replay; if the first eligible attribution step is the live due step, persisted evidence is fed into that live step without synthetic replay rows. When a same-pod reconnect pump-total delta spans both an in-flight issued dose and fallback basal exposure, runtime partitions the known pending issued-dose request from that pump total only when the request, pod identity, completion timing, and residual fallback exposure are credible; the residual is used for fallback replay and the raw pump total remains telemetry/operator-review evidence. While the pod still reports active delivery or unknown status, or no post-request pump-status refresh exists yet, the attribution remains pending and live advancement is skipped. Once a fresh settled post-request pump status exists and the original pod has not been proven replaced, absent, mismatched, non-credible (delivered units outside <code>0...requested</code>), or missing-identity delivery evidence resolves as assumed delivered per clinical policy: the requested units are recorded as the assumed delivered amount, fed into the first eligible attribution step by replay or live-step input, labeled assumed - never confirmed - in UI/telemetry, and meal-linked assumed doses schedule a meal-adaptation exclusion so algorithm meal learning does not adapt on assumed evidence. Assuming the requested amount is the conservative direction for insulin-on-board accounting (under-crediting would risk re-dosing insulin that may already have been delivered) and preserves dosing liveness. Non-credible pump-total partition still refuses to guess the fallback split: ambiguous raw	SRS-PUMP-010, SRS-PUMP-011, SRS-MEAL-012, SRS-STATE-004, SRS-STATE-005, SRS-LOG-001	TV-PUMP-009, TV-MEAL-013, TV-STATE-004, TV-STATE-005, TV-LOG-001, TV-PUMP-010	Medium (S3×P1); exact-freeze AUTO and SIM/pod-simulation execution passed. Residual risk is accepted under D07; no separate physical canceled-partial/reconciliation claim is made

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			pump total is not displayed or persisted as actual fallback residual delivery. If a different/new pod appears before reconciliation, runtime records that disposition, preserves the same algorithm session, assumes the issued dose was delivered, feeds that assumed amount into the first eligible attribution step by replay or live-step input, clears the old-pod pending attribution after consumption, scrubs unrelated replacement-pod <code>lastDelivery</code> from the resumed live algorithm input, and does not block later insulin-adding automatic commands solely because the original pod cannot be reconciled. If the operator explicitly confirms pod replacement from an unresolved meal-progress modal after the old pod is unavailable or timed out, runtime records <code>user_abandoned_unavailable_pod</code>, stores assumed-delivered evidence, clears only meal-progress UI state, and leaves attribution for replay/live-step consumption; missing original pod identity is accepted only for this assumed-delivered path when step, units, delivered bounds, and non-active pump state match. Every pump command and status await is additionally bounded by an app-level 120-second watchdog that fails the command as uncertain (or the status read as blocked) instead of stalling the serialized loop work pass; late or duplicate transport completions after the watchdog fires are ignored safely (SRS-PUMP-011).			
RA-018	Sustained loss of connected CGM data drives the algorithm to forced open-loop: automated commands are blocked and the pod's programmed basal remains masked at 0 U/hr, producing a zero-insulin window that persists until fingerstick BG entry or CGM recovery (field-observed 2026-07-10: 95 minutes)	High (S3xP2)	Algorithm-native dropout tolerance delivers an exponentially tapering micro-dose for the designed 60-minute window before forcing open loop; per-instance algorithm isolation restores and regression-pins that full window (dual-instance isolation correction); escalating alert pair (informational at the first blocked step, safety-critical after 3 consecutive blocked steps) instructs the fingerstick-BG remedy; a manual fingerstick BG re-closes the loop; per-step telemetry records the blocked window for study oversight. fingerstick-supported CGM-outage dosing controls implemented 2026-07-14 under SRS-OUTAGE-001..006 and SDD-OUTAGE-001: persisted BG-due schedule (120-minute cadence, 60 when the measured value is >200/<80 mg/dL), tiered pre-scheduled alerts (due-soon actionable, backup-basal informational, overdue safety-critical), Home countdown plus gold BG-entry emphasis for due-soon and overdue requests, backup-basal bridge (active mask cancel on connected forced-open, renewal suppression while released, pre-command re-arm on re-close, conservative failure paths), replay-based reconciliation of bridged exposure, and no maximum-outage stop (deliberate, clinically owned divergence from the iLet cutoffs; short study, close monitoring, escalating alerts, remote telemetry). The due-soon/overdue tiers are OS notifications and depend on user-granted notification permission: notification permission is	SRS-ALERT-019, SRS-RUN-007, SRS-ALG-008, SRS-OUTAGE-001, SRS-OUTAGE-002, SRS-OUTAGE-003, SRS-OUTAGE-004, SRS-OUTAGE-005, SRS-OUTAGE-006	TV-ALERT-018, TV-ALG-004, TV-OUTAGE-001, TV-OUTAGE-002, TV-OUTAGE-003	Medium (S3xP1); exact-freeze AUTO and SIM execution passed, residual risk is accepted under D07, and no separate physical outage claim is made.

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			requested once at app startup (priming); iOS silently drops the pre-scheduled tiers when permission is denied or later revoked and this baseline has no in-app permission-state indicator (gap; tracked as an identified future enhancement). Since 2026-07-16 every scheduled-notification telemetry event carries the OS <code>authorization_status</code> and an unauthorized schedule logs a cloud warning, so a denied-permission phone is visible to study monitoring even though not yet surfaced in-app - the in-app alert stack and Home countdown remain active regardless of permission, and the IFU setup instructions require confirming notifications are enabled during provisioning (information-for-safety control).			
RA-019	New Participant Reset is invoked unintentionally (wrong reset chosen, or its device-wipe scope misunderstood), erasing participant-scoped data mid-study: clinical configuration, local step telemetry and CSV export, the not-yet-uploaded cloud outbox queue, alert history, stored device-manager state, and the subject-scoped unlock material including burned counters; or asynchronous device work recreates outgoing-participant state after the wipe	Medium (S2xP2)	Reset is reachable only inside unlocked Clinical Settings (clinical-unlock gating); distinct side-by-side labeling (Same-Participant Reset vs New Participant Reset); refusal while a pod is active; guided stop-then-reset when a session is armed (no silent session stop); explicit destructive confirmation naming the erased data; telemetry drain guard: a final upload flush before the erase confirmation, an explicit consent line stating how many telemetry records remain undelivered and that resetting discards them permanently, a best-effort <code>telemetry.outbox.discarded</code> marker posted before the erasure, and sequence-continuity-preserving erasure so any discarded records stay server-detectable as a BionicScout gap (no more silent local-queue loss); CGM reset teardown cancels acquisition work and invalidates persistence callbacks/in-flight markers before deleting raw manager state, preventing late completion from recreating the prior participant's replacement episode; sign-out after the wipe makes the effect immediately visible; previously uploaded telemetry persists server-side in BionicScout (only local copies and the local queue are erased); dosing safety is unaffected because the reset never executes with an active pod or an armed session	SRS-CLIN-014	TV-CLIN-015	Low (S2xP1)
RA-020	Backup basal delivered during a connected CGM-outage window is not counted by the algorithm (discovered 2026-07-16 in live field data): while the bridge is released, the pod runs its programmed fallback schedule; without reconciliation, subsequent dosing can skew toward extra insulin, partly offset by blocked mode-2 asks that	Medium (S3xP2, pre-mitigation, human-use framing)	Interim telemetry control shipped 2026-07-16: every release window is quantified at mask re-arm. The frozen baseline contains the implemented accounting control: connected released steps credit live per-step using echoed asks plus identity-guarded pod deltas; a pod-away gap is reconciled by historical, schedule-weighted replay rather than one reconnect chunk. Shares are allocated in 0.01 U quanta by largest remainder so the C++ booked total equals the measured delta. The replay plan persists measured delta, re-anchor counter/identity, and first-row echo ask so a force-quit neither loses nor double-counts delivery. Real-C++ characterization, recorded-sequence differential, and simulated outage scenarios passed the exact-freeze SIM lane.	SRS-OUTAGE-006	TV-OUTAGE-002	Medium with implemented frozen-baseline control (S3xP1); clinical confirmation of 0.01 U quantized time positioning is recorded. Residual risk is accepted under D07; no separate physical outage claim is made

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
	the C++ carries as assumed delivery					
RA-021	An implausible subject weight (units fat-finger, migration defect, or absurd entry) reaches the dosing algorithm and scales every computed dose - weight was the one clinical input with no plausibility bound before the C++ (found by the 2026-07-17 algorithm test coverage review; all other clinical settings were double-bounded and double-tested)	Medium (S3xP2, pre-mitigation)	Save gate: configurations with weight outside 50...500 lbs are not usable for arming (SRS-CLIN-015; entry is clinical-unlock-gated); store top-clamp at 500 lbs bounds runaway scale for legacy-persisted values without ever raising a lower entry (0 stays unset and keeps first-launch setup gated; sub-plausible positives stay visible but unusable); algorithm last rail clamps consumed weight into 20...230 kg where the low clamp under-doses (safe direction) and the clamped value is telemetry-visible; range clinically confirmed 2026-08-21	SRS-CLIN-015	TV-CLIN-016	Medium (S3xP1)
RA-022	A Pregnancy Temporary Target is applied with the wrong value or interval, fails to apply when selected, persists beyond its intended end, or contaminates the safety-controller/fallback profile. The resulting primary dosing may be less or more aggressive than intended during activity, or an inappropriate temporary target may survive relaunch/profile/session/participant transitions.	High (S3xP2, pre-mitigation)	The participant must explicitly select one of two bounded targets and one of four bounded durations; activation is Pregnancy-only and requires an armed session, active Pod, and no pending fallback recovery. The subject-bound state persists an absolute expiry before UI success, is resolved independently for every algorithm row's execution time, and the first row at/after expiry returns to the permanent target without relying on notification delivery. Replacement/end/profile/session/reset dispositions are deterministic; New Participant Reset removes all state. The override is injected only through the primary set-point resolver; the safety target, q5 profile, and programmed four-rate backup schedule remain derived from the permanent Pregnancy configuration, as clinically selected 2026-07-24. Home/Settings show target and end time, a local expiration notification is scheduled, and lifecycle plus per-step telemetry distinguish activation from observed algorithm use in BionicScout.	SRS-CLIN-016, SRS-LOG-013, SRS-RUN-007	TV-CLIN-017, TV-LOG-013	Medium (S3xP1); exact-freeze AUTO execution passed. Residual risk is accepted under D07; no separate physical/live-telemetry claim is made
RA-023	The app remains bound to an ended G7 and never acquires its replacement, or releases the correct binding and adopts a nearby participant's G7, causing unavailable or wrong-person glucose to influence automated dosing.	High (S3xP2)	Settings exposes a manual replacement action; ordinary BLE search and reading staleness never clear the bound identity. Replacement intent and prior identity are synchronously offered to app storage before the lower BLE binding is released; direct manager reconstruction preserves that state; repeated manager callbacks do not restart the timer; and adoption requires authenticated glucose discovery with activation evidence. A ten-minute in-app alert directs completion in the Dexcom app and retry from Settings, and stable stall/adoption event identities reduce duplicate Scout evidence. Device-reported sensor failure/session end and remote disconnect during pending authentication may also start replacement; routine disconnect does not. The current integration has no authoritative intended-sensor handoff	SRS-CGM-006, SRS-ALERT-020, SRS-LOG-014	TV-CGM-006, TV-ALERT-019, TV-LOG-014	Medium (S3xP1), accepted for IDE submission under D06 and D07. The controlled multi-device exercise and final deployment disposition remain required before first participant deployment.

RA-ID	Hazard	Initial Risk	Key Controls	SRS Links	Verification Links	Residual Risk
			identity. The September 3 adversarial review found that reopening CGM setup during active replacement can install a fresh manager and reset episode evidence, multiple already-queued candidate callbacks have no atomic single-winner gate, automatic triggers may begin without planned physical isolation, and the IFU's post-connection comparison does not gate first-reading use by an armed runtime. No wrong-sensor adoption was observed in the supporting field event.			
RA-024	A combined recovery export pairs the retained step CSV with Algo2015 files from the wrong run or captures a changing file inconsistently, causing investigators to reconstruct an incorrect session when primary telemetry has a gap.	Medium (S2×P2)	Timestamped log/data files are grouped by parsed epoch and the fixed matrix joins only the epoch it names. The active C++ real-T0 inspection epoch selects only an exact group; an unavailable exact group produces CSV-only output, while absent inspection state permits only the newest valid grouped epoch. CSV and algorithm sources are each copied to their observed start size before ZIP creation; ZIP input is staged snapshots rather than live paths; CRC/size and byte-exact extraction are regression-tested. The Settings modal/picker path was removed, eliminating its repeated concurrent-presentation failures. The feature is read-only and cannot change dosing or algorithm output.	SRS-LOG-015, SRS-SEC-001	TV-LOG-015	Low (S2×P1); exact-freeze AUTO and scoped SEC execution passed, and residual risk is accepted under D07. No separate physical share/open claim is made.

RA-009 controlled clarification (v1.38): field evidence showed that a long debug-logging backlog could amplify itself after the chatter lane reached its cap. The strengthened control uses a protected SQLite ledger with no count eviction for retained clinical/runtime evidence, transactionally and repeatably imports earlier file spools, completes telemetry calls after durable commit rather than network drain, preserves independent retry scheduling, drains retained evidence before bounded chatter, and aggregates chatter-cap loss after transport progress. An additive bounded batch route improves catch-up throughput while unsupported-route fallback preserves the deployed single-event contract. A separate protected SQLite archive preserves the complete active-session step history for clinical export while Recent Dose Steps remains bounded. New Participant Reset physically recreates the outbox database and removes its sidecars while preserving only install-scoped sequence metadata; explicit algorithm-session reset securely purges the step archive. Byte-marker regressions verify that cleared participant content is absent from the affected SQLite artifacts. Notification-clear telemetry is emitted only for an actual persisted lifecycle transition, and Settings remains passive with no alert or dosing dependency. Storage exhaustion is now bounded by available protected app storage rather than an arbitrary clinical-entry count; an actual local commit failure cannot itself be reported to cloud while storage is unavailable. Working verification is TV-LOG-001/TV-LOG-012; live update-in-place recovery, Scout catch-up, post-deploy batch observation, and formal promotion remain required.

RA-009 / RA-024 submission-scope clarification (v1.48): clinical review on 2026-08-27 approved Scout as supportive monitoring and corroboration rather than an official study outcome or safety-report source. Official CGM outcomes come from Dexcom Clarity, insulin-delivery outcomes from

the BionicLoop exported CSV, and adverse events and other safety information from staff-entered electronic case-report forms. The device-local recovery ZIP remains primary software session-reconstruction evidence for its covered interval. This clarification changes neither the controls nor the residual scores.

RA-017 controlled clarification (v1.24): DASH can report `bolusNotDelivered = 0` after a canceled bolus and app relaunch. That register value alone is therefore not evidence of full completion. Durable cancellation evidence and legacy partial evidence remain authoritative; only a partial explicitly labeled as capped while delivery was still active may be raised by a later same-pod settled observation. This control prevents a canceled partial from being fabricated as a full delivered dose in algorithm input while preserving the existing conservative assumed-full policy when no reliable partial evidence exists.

RA-017 controlled clarification (v1.31): the 2026-07-30 field suspension incident showed that a locally cached 2.9 U request rejected by the Pod as suspended could be paired with `bolusNotDelivered = 0`, displayed as delivered, and fed to steps 21-23 because no valid issued-dose attribution existed to deduplicate it. The control now invalidates matching adapter request metadata after a definitive blocked result and strips any matching cached delivery; uncertain results retain reconciliation state. The independent idle-only meal preflight prevents the normal known-suspension path from reaching either algorithm or command construction.

Residual Risk Acceptance Notes

- RA-009: Sponsor-approved closure landed 2026-07-17. OS file sharing is absent, CSV export is sealed to the clinical-gated share path with complete protection and backup exclusion, and telemetry completeness is observable through SRS-LOG-012 sequence numbering. No separate sponsor export-posture decision remains open for the IDE baseline.
- RA-013: Offline single-use Clinical Settings unlock is accepted as an investigational single-device control; broader production authorization/role-based access remains deferred unless explicitly pulled into scope.
- RA-005: Residual resolved to `Low` (previously recorded ambiguously as `Low/Medium`): dose quantization is validated at command construction and reconciled in telemetry (TV-PUMP-003, TV-LOG-001).
- RA-015: Exact-freeze automated and simulation execution covered the implemented reconnect/fallback gate. Clinical written confirmation of RES-004 is recorded in `IDE_Clinical_Policy_Confirmation_2026-08-21.md`. Residual risk is accepted under D07; no separate physical reconnect/background claim is made.
- RA-018: The connected-outage zero-insulin window past the 60-minute dropout tolerance is tracked as ANOM-003. The clinically approved fingerstick-supported CGM-outage dosing design (2-hour fingerstick cadence, hourly when BG >200/<80; backup-basal bridge; no maximum-outage stop) is IMPLEMENTED as of 2026-07-14. Exact-freeze automated, algorithm, and simulation execution is complete, and residual risk is accepted under D07; no separate physical-device claim is made.

- RA-014, RA-015, RA-016, and RA-017: Exact-freeze execution evidence is mapped in the canonical RTM. ALG-DEV-SA-006 is closed by companion evidence. Physical hardware claims remain deferred unless signed operator evidence is completed and accepted.
- RA-018 (notification permission): the due-soon/overdue outage tiers are pre-scheduled OS notifications and do not fire if the participant has denied notification permission; no in-app permission surfacing exists in this baseline (2026-07-14 audit finding). The in-app alert stack and Home countdown are permission-independent, and the IFU setup checklist requires enabling notifications at provisioning. Accepted for the investigational study given provisioning verification and close study-team monitoring; in-app permission surfacing is an identified future enhancement.
- RA-023: No Dexcom-app handoff identity is exposed by the current integration. The accepted no-handoff design uses explicit replacement acquisition followed by authenticated first-sensor discovery and trained verification. Under D06, the listed multi-device bench test is deferred from submission assembly until before first participant deployment. The provisional S3xP1 rating is accepted for submission; the exercise result controls final deployment disposition.

Residual Probability Rationale

Seventeen register rows carry a **Medium** ($S3 \times P1$) residual. That is a consequence of the scale, not a default: $S3$ reflects that each of those hazards, unmitigated, has a plausible path to incorrect or unaccounted insulin delivery or a missed safety-critical alert, and $P1$ is claimed only where the listed controls are implemented and linked to passing verification and the residual path to harm requires multiple independent failures. Per-row discriminators an auditor should expect:

- RA-001 initial $P3$ (iOS wake variability occurs routinely absent controls) versus RA-018 initial $P2$ (a sustained multi-hour connected CGM outage was observed once in field use): the triggers have different base rates and are deliberately rated differently.
- RA-010 initial equals residual ($S2 \times P2$): UI-state controls bound the consequence and improve detectability, but occasional workflow slips remain expected with any UI (human-factors floor). The row is acceptable at **Medium** under the criteria; it is not driven to $P1$.
- RA-008 residual lands $P2$, not $P1$: telemetry gaps remain plausible (storage pressure, upload interruption) even with always-on capture; $S2$ severity bounds the row at **Medium**.
- RA-014 residual $P1$ is claimed under the documented probability definition (occurrence of harm, not of the triggering defect): The dual-instance isolation incident demonstrated the independent safeguard layer containing a real latent-defect escape without participant harm. Exact-freeze behavioral verification and the linkage companion passed; residual risk is accepted under D07.
- RA-023 residual $P1$ is claimed provisionally because ordinary staleness cannot release the bound sensor, replacement requires an explicit or defined device-lifecycle trigger, and adoption requires authenticated glucose discovery with trained post-connection verification. A path to wrong-person dosing therefore requires a replacement episode, a nearby non-intended candidate, failure of the intended-device workflow, and failure of the trained verification control. Supporting verification is incomplete for CGM-screen reentry, concurrent candidates, automatic-trigger isolation, and first-reading use. D06 and D07 accept $S3 \times P1$ for submission, subject to the controlled exercise and final deployment disposition before first participant deployment.
- Rows whose controls eliminate the hazard mechanism (RA-002 sanitization, RA-005 quantization validation, RA-007 refresh wiring) reach $P1$ /Low directly; rows whose controls interpose barriers (RA-003, RA-004, RA-006, RA-011, RA-012, RA-013, RA-015, RA-016, RA-017) claim $P1$ because harm additionally requires the in-app escalating-alert control to fail during a short, closely monitored study.

Overall Residual Risk Conclusion (Benefit-Risk)

With the listed controls implemented and linked to verification, no register row carries a **High** residual; RA-009 is closed at Low under the sponsor-approved sealed-export and provable-delivery posture. Exact-freeze execution is complete: STP-ALG-001 passed behaviorally and its

6/6 linkage companion closes ALG-DEV-SA-006; STP-AUTO-001 completed and its corrected 44/44 UI companion closes AUTO-DEV-001 and AUTO-DEV-002; STP-SIM-001 passed; and TV-SEC-001 passed 68/68 scoped controls. All 24 IDE-scope risk-analysis rows have execution evidence mapped. The anticipated benefit of investigational closed-loop dosing justifies the documented residual risks. Final benefit-risk acceptance is recorded under D07; the accepted D06 multi-G7 exercise remains a predeployment condition.

Software Verification and Validation Plan

(SVVP)

Status: Frozen-baseline verification plan and evidence approved for IDE submission Version: 1.91
 Owner: BionicLoop engineering Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, Build 843 evidence acceptance and final software-package approval, controlled email dated 2026-09-07 EDT Baseline freeze SHA:
 91c0e98a9bc9429a0486bebdebf fc7d8dbbe300e Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled verification draft
0.9	2026-04-06	BionicLoop engineering	Added document-control metadata and clarified the in-scope local security verification row
0.91	2026-04-08	BionicLoop engineering	Added fallback-event persistence and Recent Dose Steps review-state coverage
0.92	2026-04-08	BionicLoop engineering	Added pump basal-schedule interface coverage for masked offline fallback
0.93	2026-04-08	BionicLoop engineering	Added blocking coverage for disruptive programmed basal-schedule replacement during active bolus and active temp-basal states
0.94	2026-04-08	BionicLoop engineering	Added masked-fallback maintenance/disarm verification rows, active-session gating coverage, and restore-event review coverage
0.95	2026-04-08	BionicLoop engineering	Clarified 20-minute renewal-window verification, deferred-maintenance coverage, and pre-step masked-fallback renewal expectations
0.96	2026-04-14	BionicLoop engineering	Added verification rows and evidence notes for reconciliation-required blocked state after offline mask expiry
0.97	2026-04-14	BionicLoop engineering	Added reconnect restore/disarm retry and explicit fresh re-arm verification notes for masked-fallback recovery
0.98	2026-04-14	BionicLoop engineering	Updated masked-fallback reconnect-recovery verification notes for fresh-session auto-arm and preserved fallback review history
0.99	2026-04-14	BionicLoop engineering	Updated masked-fallback reconnect-recovery verification notes for same-session unreconciled resume on the current due slot

Version	Date	Author	Summary of Changes
			after successful restore
1.00	2026-04-15	BionicLoop engineering	Added basal-only reconnect evidence, modeled-vs-pump-reported fallback review, and pre-step missing-fallback arm verification notes
1.01	2026-04-15	BionicLoop engineering	Added confirmed/corrected basal-only reconnect replay verification notes, coordinator replay tests, and replay-specific Home timeline review coverage
1.02	2026-04-15	BionicLoop engineering	Added dedicated cloud fallback telemetry verification for structured <code>loop.fallback.event</code> emission, duplicate suppression, and backend-facing payload mapping
1.03	2026-04-15	BionicLoop engineering	Added verification coverage for replayed-step propagation into local per-step telemetry / CSV export with explicit replay markers
1.04	2026-04-27	BionicLoop engineering	Added verification coverage that missing pump-reported delivered-insulin delta prevents fallback replay and keeps modeled expected delivery logging-only
1.05	2026-05-06	BionicLoop engineering	Added q5 nominal-basal profile, safety-track four-bucket fallback schedule, schedule-aware exposure, and fallback profile/schedule telemetry verification coverage
1.06	2026-06-02	BionicLoop engineering	Updated masked-fallback verification to pump-delta missed-step algorithm replay and added repeated no-active-pod alert coverage
1.07	2026-06-08	BionicLoop engineering	Added verification coverage for persisted-schedule-weighted pump-delta replay allocation, zero-weight suppression, and zero-delta replay rows
1.08	2026-06-12	BionicLoop engineering	Added verification coverage for generalized issued-dose attribution, replay, unresolved block, and different/new-pod no-replay disposition
1.09	2026-06-13	BionicLoop engineering	Added verification coverage for issued-dose delivered-unit credibility, replacement-pod live-step input scrubbing, and automatic resume blocking under active holds
1.10	2026-06-15	BionicLoop engineering	Updated different/new-pod issued-dose verification to assumed-delivered replay/live attribution, nonblocking replacement-pod dosing, legacy hold clearance, and meal progress modal resolution
1.11	2026-06-17	BionicLoop engineering	Added verification coverage for user-confirmed pod replacement from unresolved meal delivery progress with assumed-delivered evidence and non-meal suppression

Version	Date	Author	Summary of Changes
1.12	2026-06-18	BionicLoop engineering	Added verification coverage for retired/expired/no-active-pod fallback recovery using assumed modeled fallback exposure when pump-counter evidence is no longer recoverable.
1.13	2026-06-23	BionicLoop engineering	Added provisional pod-simulation scenario coverage for issued-dose live attribution, lost final meal response before refresh, replacement-pod assumed-delivered replay, stale cached idle rejection, meal/fallback partitioning, stale fallback evidence suppression, and schedule-weighted fallback replay allocation.
1.14	2026-06-23	BionicLoop engineering	Expanded pod-simulation coverage for canceled and consecutive-canceled meal evidence feeding subsequent meal announcements, user-escaped unavailable-pod assumed evidence, fallback-maintenance command blocking before live bolus delivery, and relaunch fallback replay with unavailable CGM.
1.15	2026-06-23	BionicLoop engineering	Added explicit pod-simulation coverage for unresolved correction-only and basal-only issued doses without recoverable pod identity.
1.16	2026-06-23	BionicLoop engineering	Added pod-simulation coordinator coverage for ambiguous fallback restore without replay or modeled-dose injection.
1.17	2026-06-23	BionicLoop engineering	Added pod-simulation coverage for assumed old-pod fallback replay, missing-status/nonreplayable fallback-plan clearing, and matching meal-progress cleanup after issued-dose evidence consumption.
1.18	2026-06-23	BionicLoop engineering	Added pod-simulation coverage for reservoir-capped partial meal delivery feeding actual delivered units without replay or unresolved meal-progress state.
1.19	2026-06-23	BionicLoop engineering	Added pod-simulation coverage for reservoir-capped fallback replay using observed pump delta instead of modeled exposure.
1.20	2026-06-23	BionicLoop engineering	Added pod-simulation coordinator coverage for meal/fallback overlap partition replay with the meal dose merged into the first fallback-active replay row.
1.21	2026-06-23	BionicLoop engineering	Added app-layer recent-dose persistence/display verification for merged meal/fallback replay evidence source, disposition, request step, requested units, and delivered units.
1.22	2026-06-23	BionicLoop engineering	Added local step CSV export verification for merged replay evidence source, disposition, and failure-reason fields.
1.23	2026-06-23	BionicLoop engineering	Added runtime cloud step-event emission verification for merged replay evidence source, disposition, request-step, requested-unit, and delivered-unit fields.
1.24	2026-07-08	BionicLoop engineering	Added reference-host M4 no-boundary-cancel liveness-bound coverage for DASH bolus completion, disconnect/unknown

Version	Date	Author	Summary of Changes
			delivery state, and fresh idle issued-dose consumption.
1.25	2026-07-08	BionicLoop engineering	Replaced merged-overlap issued-dose coverage with pre-fallback confirm row + sentinel + seam-echo coverage (incident regression, wider-gap sentinels, echoed-ask no-room merge) plus real-C++ characterization of the bug shape, fixed shapes, and the full confirm/sentinel/fallback sequence.
1.26	2026-07-09	BionicLoop engineering	Replaced carousel-navigation coverage with Home alert-stack root-cause suppression coverage (generic-alert hiding, safety-critical never suppressed, unrelated untouched).
1.27	2026-07-09	BionicLoop engineering	Added open-composer revalidation coverage: in-place re-arm, inline fresh-reason blocked messaging, cancel-delivery precedence, continue-on-available.
1.28	2026-07-09	BionicLoop engineering	Added simulated-preview meal-progress exemption coverage (runtime auto-resolution never dismisses UI-test/IFU-capture preview progress; production resolution semantics unchanged).
1.29	2026-07-09	BionicLoop engineering	Added hidden-related-alert count coverage for the Home alert stack (SRS-ALERT-011 amendment: suppressed alerts must be counted and pointed at Alert Center).
1.30	2026-07-09	BionicLoop engineering	Corrected legacy evidence citations to the controlled working-evidence location and clarified that formal closure requires execution in the formal lane at the freeze SHA with required metadata.
1.31	2026-07-10	BionicLoop engineering	Added dual-instance isolation coverage: primary and safety algorithm copies share no state, and the 60-minute CGM-dropout window is pinned against the real C++.
1.32	2026-07-10	BionicLoop engineering	Independent review aligned TV-ALG-004 with the dual-instance isolation contract and SRS-ALG-008, corrected coverage placement and broken family links, aligned TV-ALERT-010 with the alert stack, aligned TV-SEC-001 with the RA-009 investigational export posture, and removed obsolete feasibility wording.
1.33	2026-07-14	BionicLoop engineering	Added fingerstick-BG dropout-clock characterization coverage (real C++): BG clears forced-open and restarts the full 60-minute window; design input for the CGM-outage BG-run policy (ANOM-003/RA-018).
1.34	2026-07-14	BionicLoop engineering	Added fingerstick-supported CGM-outage verification rows TV-OUTAGE-001..003 and pump-command-watchdog row TV-PUMP-010. Formal-lane execution was deferred to the subsequent freeze candidate.
1.35	2026-07-14	BionicLoop engineering	Clinical-feedback rows: TV-CLIN-014 (first-launch defaults Standard/120/75%/40) and TV-CLIN-015 (New Participant Reset store-clearing matrix + armed/active-pod refusal), covered by

Version	Date	Author	Summary of Changes
			BionicLoopClinicalSettingsRuntimeTests and BionicLoopRuntimeEngineSessionInfrastructureTests.
1.36	2026-07-14	BionicLoop engineering	TV-CLIN-014/015 coverage extended: empty-store defaults-without-persist and legacy-migration-preserved pins, per-mode wizard target defaults, and post-reset first-launch-gate assertions (BionicLoopClinicalSettingsRuntimeTests, BionicLoopRuntimeEngineSessionInfrastructureTests).
1.37	2026-07-14	BionicLoop engineering	Independent review corrected stale test citations, aligned TV-CLIN-015 with guided stop-then-reset, clarified that TV-OUTAGE-002 uses existing fallback-replay evidence, and aligned reset terminology.
1.38	2026-07-15	BionicLoop engineering	TV-LOG-010 added (build-designated cloud-log profile: parse + launch-reconcile pins in the cloud-log policy and session infrastructure test classes); Scout-side ingest coverage noted (CT_J5_013 pins session-metadata preservation and app_version/build_number on the CloudWatch line).
1.39	2026-07-15	BionicLoop engineering	Added STP-INT-001 formative real-device integration exercises with machine-checkable signatures, telemetry and application-log evaluation, and supporting working-evidence bundles. Checker behavior was validated for empty windows and positive signature matching.
1.40	2026-07-16	BionicLoop engineering	TV-LOG-011 added (remote logging config parse/apply/precedence) and TV-LOG-004 reworked to the no-in-app-control posture; working-coverage bullet for the 2026-07-16 diagnostics additions (scheduler authorization-status enrichment, watchdog timeout observer, Scout route tests).
1.41	2026-07-16	BionicLoop engineering	TV-LOG-011 extended for the SRS-LOG-011 v1.34 amendment: periodic re-fetch loop coverage and the read-only Settings effective-state status text (precedence + source attribution), with the three new test names listed.
1.42	2026-07-16	BionicLoop engineering	SRS-LOG-010 supersession: TV-LOG-010 rescinded (profile/session tests deleted with the code), TV-LOG-011 re-stated for the sole-lever posture (legacy-storage-ignored, logger-honors-remote, change-observer, renamed status test), build-profile working-coverage bullet retired.
1.43	2026-07-16	BionicLoop engineering	TV-UI-011 added (SRS-UI-009 server-configurable investigational badge): captured-fixture parse pins, three-state effectiveText, refresh semantics, shared periodic loop over both server configs; Scout-side route/panel/badge coverage noted.
1.44	2026-07-16	BionicLoop engineering	TV-OUTAGE-003 extended for the SRS-OUTAGE-002 v1.37 amendment: outage-active gate truth table pinned (offline mode / relaunch blip / fingerstick / mid-BG-run / recovery / skip).

Version	Date	Author	Summary of Changes
1.45	2026-07-16	BionicLoop engineering	Extended TV-OUTAGE-002 for SRS-OUTAGE-006 v1.38 release-window quantification under RA-020.
1.46	2026-07-17	BionicLoop engineering	TV-CLIN-016 added (SRS-CLIN-015 weight plausibility bounds): save-gate range, store top-clamp semantics, and algorithm last-rail clamp pinned by BionicLoopClinicalSettingsSavePolicyRuntimeTests, BionicLoopClinicalConfigStoreRuntimeTests, and RealBUDosingAlgorithmStateTests.
1.47	2026-07-17	BionicLoop engineering	RA-009 closure: TV-SEC-001 rewritten to the implemented sealed-export controls with concrete test symbols; TV-LOG-012 added for sequence-numbered envelopes (SRS-LOG-012; renumbered same-day from a duplicate -011 assignment).
1.48	2026-07-17	BionicLoop engineering	Coverage review corrected stale test-class and method citations in the deferred-validation block and TV-MEAL-009.
1.461	2026-07-17	BionicLoop engineering	Extended TV-CLIN-015 for the SRS-CLIN-014 telemetry drain guard: outbox counts, drain-then-count behavior, discard marker and erasure, discard-before-wipe ordering, confirmation copy, and upload-rejection alert lifecycle. (Renumbered 2026-07-21 from duplicate 1.46.)
1.49	2026-07-18	BionicLoop engineering	TV-OUTAGE-002 extended for the SRS-OUTAGE-006 v1.42 amendment (2026-07-18 field incident): released-window pod-away gap reconciliation via the standard bounded replay - seven scenario-suite pins covering crossover, both-links-down horizon behavior, force-quit relaunch seams, identity mismatch, zero-delta retraction replay, and insulin conservation; also added the measured real-C++ chunk-vs-replay divergence suite and the released-outage classifier-parity test.
1.50	2026-07-18	BionicLoop engineering	Extended TV-OUTAGE-002 after independent safety review with mid-save termination recovery, drift-skip re-synthesis, non-flat quantized weighting, profile-timezone weighting, and real-C++ ledger-conservation checks; added the Algo2015 diagnostic-artifact protection regression to TV-SEC-001.
1.51	2026-07-18	BionicLoop engineering	TV-ALERT-017 extended to pin the revised check-BG body copy and Home presentation policy: no inline <code>Enter BG</code> action, separate Home BG-entry emphasis retained, and unrelated login primary action preserved.
1.52	2026-07-19	BionicLoop engineering	TV-MEAL-002 and TV-ALERT-002/010 extended for exact-step pending-BG meal execution across primary/safety tracks, real-C++ forced-open recovery with combined BG + meal input, invalid/stale/pump-unavailable negative cases, and pump-signal precedence over same-severity check-BG without overriding safety-critical alerts.

Version	Date	Author	Summary of Changes
1.53	2026-07-20	BionicLoop engineering	TV-BG-008/UI-006 extended for spatially separate exact-value BG review, value-preserving Change, and rapid-repeat-tap protection; TV-ALERT-017/OUTAGE-003 extended for direct fingerstick-request Home emphasis policy.
1.54	2026-07-20	BionicLoop engineering	TV-ALERT-017 extended with a rendering-regression pin proving the repeating manual-BG request emphasis changes only the localized halo while card-shadow properties remain static, including the static Reduce Motion presentation.
1.55	2026-07-20	BionicLoop engineering	TV-MEAL-007 extended for the SRS-MEAL-006 v1.49 consumed-carve-out amendment: decision-level pins for the consumed-carve-out inline message vs the generic backup-basal copy across both routing branches, unreconciled-issued-dose non-repaint, end-to-end pure pins for the re-closed-loop re-arm and persisting-forced-open sub-shapes (including the <code>backupBasalRunning</code> submit-availability passthrough), recovery re-arm/continue rows, and the carve-out capture/consumption helper truth tables (ten new test names listed in the TV-MEAL-007 coverage bullet).
1.56	2026-07-20	BionicLoop engineering	TV-MEAL-007 reworked for the SRS-MEAL-006 v1.50 sponsor correction: the consumed-carve-out rows now pin the transient waiting hold (passive applying notice, composer armed) while BG consumption is in flight across both routing branches, the generic backup-basal fallback once the consuming step has published, the in-flight helper truth table (cleared value plus unpublished captured target), the submit-time hold predicate, and the unchanged recovery/unreconciled pins (twelve test names listed in the TV-MEAL-007 coverage bullet, replacing the v1.49 list).
1.57	2026-07-20	BionicLoop engineering	Extended TV-PUMP-009 after a force-quit mid-bolus field incident: fresh same-pod post-completion evidence may upgrade capped in-progress or assumed evidence, while canceled partial, early-read, active-delivery, and different-pod cases remain guarded. (Renumbered 2026-07-21 from duplicate 1.55.)
1.58	2026-07-21	BionicLoop engineering	TV-LOG-009 extended for SRS-LOG-009 v1.51 (steady-cadence fallback schedule visibility): unchanged-check lean payload, per-bucket observed-slot counts, and Recent Dose Steps outcome-label/provenance rows pinned by <code>testUnchangedScheduleCheckEventCarriesScheduleAndProfileCoverageWithoutBulkRates</code> , <code>testArmedEventCarriesPerBucketObservedSlotCounts</code> , <code>testScheduleRefreshEventCarriesFullProfilePayloadWithPerBucketCounts</code> (new <code>BionicLoopFallbackSchedulePayloadTests</code>), <code>testRecentDoseTimelineFallbackScheduleRowsShowOutc</code>

Version	Date	Author	Summary of Changes
			omeLabels, testRecentDoseTimelineFallbackUnchangedRowShowsScheduleSegmentsCoverageAndProvenance, testRecentDoseTimelineFallbackProvenanceMarksAllImputedAndSingleObservedBuckets (new BionicLoopRecentDoseFallbackProvenanceTests), and the extended testCloudTelemetryReporterSendsStructuredFallbackEventPayload pin; core per-bucket truth table in Q5NominalBasalProfileTests (testObservedSlotCountsByBucketAreAllZeroForEmptyProfile, testFourBucketScheduleCarriesPerBucketObservedSlotCounts, testFullyObservedDayReportsFullPerBucketObservedSlotCounts). (Corrected 2026-07-21 from erroneous 1.47.)
1.59	2026-07-21	BionicLoop engineering	TV-PUMP-009 fresh-completion safety regression expanded: exact cancel + relaunch + zero bolusNotDelivered adapter and engine shapes remain partial; explicit in-progress provenance still permits completion upgrade; cancellation provenance is labeled; legacy evidence without the optional context decodes and remains conservative. TV-ALERT-017 truth-synced to the accepted centered shadow bloom and no-glow Reduce Motion behavior.
1.60	2026-07-21	BionicLoop engineering	TV-MEAL-002 expanded for the borrowed-step forced-open transition: pure boundary tests pin blind steps 12/13, app availability pins prospective blocking and fresh-CGM/exact-BG exceptions, coordinator tests prove rejection occurs before meal persistence/algorithm/pump activity, and outage-context tests pin the qualifying-step anchor plus legacy decode compatibility.
1.61	2026-07-22	BionicLoop engineering	TV-MEAL-002 extended with connected-active-delivery precedence regressions: live .delivering status must map to pumpDelivering over pending meal/issued-dose attribution, while idle unresolved attribution keeps reconciliation guidance.
1.62	2026-07-22	BionicLoop engineering	TV-MEAL-007 extended for the intermediate-publication field regression: helper truth-table coverage pins lastExecutedStep == capturedTarget with a lagging qualifying-glucose marker as still in flight, final marker publication as complete, and a pump-delivering revalidation as replacing the transient BG/backup-basal state with connected busy-pump guidance.
1.63	2026-07-24	BionicLoop engineering	Added TV-CLIN-017/018 and TV-LOG-013 for Temporary Target lifecycle, primary/safety separation, replay-time target resolution, participant UI/notification behavior, Standard-to-Pregnancy meal defaulting, app telemetry, and BionicScout

Version	Date	Author	Summary of Changes
			projection. Physical-device and formal evidence remain pending.
1.64	2026-07-28	BionicLoop engineering	Revised STP-INT-001 for the next formative field round: retained INT-01..10, added Temporary Target exercises INT-11/12, added pre-distribution safety checks, and required exact install identity plus signature calibration.
1.65	2026-07-30	BionicLoop engineering	Extended TV-CLIN-017 and alert coverage for direct suspend/resume ownership, duplicate suppression, delivery-state policy, reminders, and alert routing; added saline-Pod field drill INT-13.
1.66	2026-07-30	BionicLoop engineering	Extended TV-SEC-008 with persistent signed-out Home login access, actionable Alert Center and Account & Session routes, login-specific notification routing, and UI tests that preserve the active local-therapy path while returning to authentication.
1.67	2026-07-30	BionicLoop engineering	Extended TV-CLIN-017 with target-only and duration-only draft-loss warnings, explicit keep/discard behavior, Settings ordering, participant-reset-copy removal, and app-standard Temporary Target selector-state verification.
1.68	2026-07-30	BionicLoop engineering	Extended TV-UI-005 with a scroll-position regression proving Manual BG and Meal Announcement remain visible, hittable, and fixed while an expanded multi-alert Home surface scrolls.
1.69	2026-07-30	BionicLoop engineering	Extended TV-LOG-012 for permanent-rejection structured evidence, no generic participant alert, Subject ID Conflict alert isolation, retryable 429 behavior, backward-compatible outbox decode, and one-time launch migration of earlier-build generic alerts/notifications; removed the superseded alert-latch claim from TV-CLIN-015.
1.70	2026-07-30	BionicLoop engineering	Extended TV-CLIN-017 and alert verification for direct Resume Insulin action, duplicate-command suppression, retryable failure, pump-confirmed alert clearing, exact 15-minute severity escalation, relaunch recovery, non-downgrade on repeated source issue, and fresh safety-critical notification.
1.71	2026-07-30	BionicLoop engineering	Extended TV-MEAL-002/TV-PUMP-009 and UI coverage for suspension preflight, no algorithm/cadence/persistence side effects, immediate post-resume availability, matching-only definitive-block cache cleanup, uncertain-outcome retention, paused Home status, and the field-observed step-20 fabricated-feedback regression.
1.72	2026-07-30	BionicLoop engineering	Added TV-CGM-006, TV-ALERT-019, and TV-LOG-014 for explicit/persistent G7 replacement acquisition, staleness non-adoption, timed guidance, alert routing/clear behavior, and durably deduplicated stall/adoption telemetry.

Version	Date	Author	Summary of Changes
1.73	2026-07-31	BionicLoop engineering	Extended TV-ALERT-013 with the July 31 suspension/fallback-recovery field sequence, stale blocker replacement, successful-step clearing, single-alert content transition, and suspend-ended Home precedence; extended TV-CLIN-015 with asynchronous G7 acquisition persistence/telemetry invalidation during New Participant Reset.
1.74	2026-08-06	BionicLoop engineering	Extended TV-LOG-001 with Recent Dose Steps verification for fingerstick provenance, exact manual BG value, concurrent CGM retention, and absent-sentinel behavior.
1.75	2026-08-11	BionicLoop engineering	Added TV-ALG-012 for reference-host total-insulin command construction and accepted next-step reconciliation of combined meal/basal/bolus output.
1.76	2026-08-11	BionicLoop engineering	Extended TV-CLIN-001 to verify the initial-configuration unlock exception closes after first save and that later Clinical Settings access/edits remain offline-unlock gated.
1.77	2026-08-13	BionicLoop engineering	Extended TV-LOG-012 with the August telemetry-backlog incident regressions: lossless v1-to-v2 migration, incremental writes, retained-first drain, coalesced chatter-drop markers, passive backlog status, and relaunch-stable notification-clear deduplication.
1.78	2026-08-17	BionicLoop engineering	Extended TV-LOG-001/012 for complete active-session step archiving and SQLite telemetry-ledger migration, uncapped retained evidence, durable-enqueue/network separation, retry liveness, 10,000-row exact-once load, and backward-compatible batch ingest.
1.79	2026-08-20	BionicLoop engineering	Added TV-BG-013 for the persisted pending-to-used fingerstick chart transition, original-time stability, same-step deduplication, and omission of invalid or discarded pending state.
1.80	2026-08-20	BionicLoop engineering	Added TV-LOG-015 for session-grouped, byte-faithful legacy Algo2015 artifact export, including partial/empty sets, live-file append boundaries, matrix-file association, ZIP naming, and presentation text.
1.81	2026-08-20	BionicLoop engineering	Extended TV-LOG-015 with stable single-presentation transition coverage across session selection, confirmation, preparation, share readiness, failure, and empty inventory.
1.82	2026-08-20	BionicLoop engineering	Replaced TV-LOG-015's obsolete Settings presentation tests with combined CSV/algorithm ZIP, exact-session isolation, CSV-only mismatch, and bounded CSV snapshot coverage through the existing Recent Dose Steps share path.
1.83	2026-08-20	BionicLoop engineering	Extended TV-ALERT-013 with fresh-idle-after-unavailable recovery copy, stale/non-idle/no-Pod rejection, no-work status

Version	Date	Author	Summary of Changes
			refresh, relaunch retention, unknown-state regression, and next-success clearing.
1.84	2026-08-20	BionicLoop engineering	Extended TV-UI-010 with a point-only CGM presentation regression proving the connector stroke and connected area fill remain disabled.
1.85	2026-08-21	BionicLoop engineering	Bound the SVVP to the immutable 2026-08-21 software freeze, recorded that primary execution is complete while approval remains pending, and replaced workstation-specific script paths with controlled identifiers. No verification method or result changed.
1.86	2026-08-21	BionicLoop engineering	Added candidate dependency-control characterization and verification for the exact CryptoSwift 1.10.0 requirement and tracked app-workspace resolution. Recorded the passing working results without changing or promoting the exact-freeze TV-SEC-001 evidence.
1.87	2026-08-27	BionicLoop engineering	Replaced internal development labels, test-device shorthand, and colloquial field-test terminology with reviewer-neutral descriptions and updated the approval role. No verification method or result changed.
1.88	2026-08-27	BionicLoop engineering	Aligned the deferred cloud-verification condition with the approved supportive Scout scope. No verification method, result, or product behavior changed.
1.89	2026-08-28	BionicLoop engineering	Replaced stale draft/proposed/provisional labels with the controlled protocol and implemented simulation status, added the exact-freeze execution boundary, and corrected the canonical serial UI/app commands and requirement-family markup. No verification method, result, or product behavior changed.
1.90	2026-09-03	Software Developer	Recorded the post-freeze RA-023 coverage limitation: the passing G7 acquisition tests cover direct manager restoration and a single candidate, but not CGM-screen setup reentry, concurrent candidates, automatic-trigger isolation, or first-reading use before staff comparison. The exact-freeze result is unchanged; these scenarios remain open under D06.
1.91	2026-09-03	Software Developer	Aligned Build 843 designation, exact-freeze coverage, test identifiers, deferred claims, and controlled evidence references with the reviewer-style package audit. No test method, result, or frozen software behavior changed.

The interim identifier 1.461 preserves the original correction of a duplicate 1.46 entry and is not a chronological renumbering error.

Working verification for revision 1.60 is pinned by `MealAnnouncementOfflinePolicyTests`, `BionicLoopMealAnnouncementOfflineAvailabilityTests`, `LoopRuntimeCoordinatorMealExecutionTests.testMealAnnounceBlocksBorrowIntoProspectiveThirteenthBlindStepBeforePersistence`, `...testMealAnnounceAllowsBorrowIntoTwelfthBlindStep`, `...testExactStepManualBGAllowsMealOnProspectiveThirteenthBlindStep`, `...testFreshCGMAllowsMealOnProspectiveThirteenthBlindStep`, `OutageBGRunPolicyTests.testContextRoundTripsThroughCodable`, `...testContextDecodesLegacyStateWithoutQualifyingGlucoseStep`, and `BionicLoopRuntimeEngineOutageBGRunTests.testAcceptedCGMReanchorsTheSchedule / ...testAcceptedFingerstickReanchorsWithEscalationForHighValue`.

1. Test Document Acronyms

Common structure used here:

- SVVP: Software Verification and Validation Plan
- STP: Software Test Protocol (test procedures and expected results)
- STR: Software Test Report (actual execution evidence)

2. Verification Strategy

Verification is split into:

- Unit tests (core logic, algorithm mapping, policy gates)
- Integration tests (runtime + adapters + persistence)
- System/manual tests (real-device behavior, BLE reconnection, onboarding flows)

Controlled protocol and reporting set:

- STP-ALG-001
- STP-AUTO-001
- STP-SIM-001
- STP-HW-001
- STP-ALERT-001
- STP-TV-Ownership-Map
- STR-Execution-and-Reporting-Guide

Submission-scope note:

- Device-to-cloud / BionicScout verification is not a relied-upon current submission claim and remains outside the controlled verification set unless submission scope is explicitly revised.

- For the current software package, TV-SEC-001 remains the only in-scope security verification row; TV-SEC-002 . . 008 are deferred from claimed closure in this pass.

3. Test Environments

- iOS Simulator for deterministic unit/integration tests.
- Physical iPhone + Dexcom G7 + OmniPod DASH for formative use and any retained connection, cadence, or physical-delivery claim. The current package does not retain a separate formal hardware-validation-study claim.

4. Entry and Exit Criteria

Entry:

- SRS and SDD IDs updated for proposed change.
- Risk impacts reviewed for affected paths.

Exit:

- All planned TV-* tests pass or deviations documented.
- Traceability matrix updated with evidence links (STR-* artifacts, logs, screenshots).
- No unresolved High severity regressions.

4.1 Exact-Freeze Execution Boundary

Lane	Current result
STP- ALG- 001	Behavioral verification passed; the 6/6 exact-freeze linkage companion closed the documentary deviation
STP- AUTO- 001	997 app tests: 996 passed, one intentional IFU reference-table generation helper skip gated by an export directory, zero failed; the corrected serial UI companion passed 44/44 and closed the three original harness deviations
STP- SIM- 001	Core, Pod, alert, and real-engine simulation lanes passed
TV- SEC- 001	68/68 scoped local controls passed

The Formal Evidence Index and Freeze Execution Report govern execution details and evidence classification. Working or formative records are not promoted by inclusion in this plan.

5. Seed Test Inventory

Test ID	Level	Purpose	SRS Link
TV-RUN-001	Unit	Expected step math anchored to first successful run, including one-step algorithm/runtime cadence reconciliation without schedule re-anchor	SRS-RUN-001, SRS-RUN-002
TV-RUN-002	Integration	Duplicate step prevention (<code>stepNotDue</code>), live executed-step/request-step alignment with algorithm input <code>stepTime</code> after cadence reconciliation, single-flight FIFO work-pass serialization with start-time state loads, and monotonic same-anchor step-base persistence (superseded stale passes abort benignly instead of regressing the base and tripping a false drift-block)	SRS-RUN-002
TV-RUN-003	Unit/Integration	Runtime <code>doWork</code> dispatch is constrained to allowed wake causes (<code>cgmUpdate</code> , <code>bgCheck</code> , <code>mealAnnounce</code> ,	SRS-RUN-003

Test ID	Level	Purpose	SRS Link
		guarded pumpReconnect)	
TV-RUN-004	Unit/Integration	Reconnect fallback executes only after an anchored session exists and only when accepted CGM receipt age exceeds the approved fallback freshness limit	SRS-RUN-004, SRS-CGM-005
TV-RUN-005	Unit/Integration	Reconnect fallback does not execute step 0, does not re-anchor cadence, and does not replay multiple missed slots	SRS-RUN-001, SRS-RUN-002, SRS-RUN-005
TV-RUN-006	Unit/Integration	Fresh accepted CGM receipt suppresses reconnect fallback and same-slot CGM/reconnect triggers coalesce to one execution	SRS-RUN-002, SRS-RUN-004, SRS-RUN-005
TV-RUN-007	System/Hardware	Real-device reconnect fallback validates current-due-step execution after CGM interruption without duplicate command application	SRS-RUN-004, SRS-RUN-005, SRS-CGM-005
TV-RUN-008	Unit/Integration	Masked fallback does not arm or refresh without an active algorithm session, arms after the current step first	SRS-RUN-006, SRS-RUN-007, SRS-PUMP-006, SRS-PUMP-008

Test ID	Level	Purpose	SRS Link
		computes a valid fallback candidate but before applying that same step's pump command when refreshed pump status is known/available, skips that pre-command maintenance hook when pump status is unavailable/unknown, arms before the next loop execution when a valid candidate was already persisted and no fallback is currently programmed, derives the programmed fallback schedule from the secondary/safety q5 nominal-basal profile when available, renews an existing mask only inside the 20-minute renewal window, suppresses same-cycle post-execution retry after a pre-command fallback event, suppresses ordinary loop execution after offline mask expiry until reconciliation-required recovery	

Test ID	Level	Purpose	SRS Link
		is resolved, re-schedules a reconnect recovery attempt from wakes blocked by the pending recovery state, requeues recovery triggers that arrive while an attempt is in flight, bounds pump-evidence exposure windows to the pod-total measurement time with post-restore totals preferred, degrades stale-baseline amount-fitting deltas to corrected pump-anchored replay, raises a safety-critical alert when an unreconciled resume may drop nonzero fallback insulin, realigns an ahead algorithm counter as the adopted step base without re-anchor and resumes dosing automatically at the adopted counter with an informational auto-clearing note (no dosing block, no manual reset demand), and holds the conservation sweep property (every recovery	

Test ID	Level	Purpose	SRS Link
		outcome replays or alerts)	
TV-ALG-001	Unit (Bridge)	Bridge null-pointer guards and edge-state reset behavior	SRS-ALG-003
TV-ALG-002	Unit (Bridge)	Input mapping and sentinel behavior (requestTime, pump availability, subject-id boundaries)	SRS-ALG-003
TV-ALG-003	Unit (Bridge)	Output/state handoff and step increment continuity at bridge boundary	SRS-ALG-003, SRS-ALG-004
TV-ALG-004	Unit (Algorithm)	Deterministic nominal golden-vector replay, plus per-instance state-isolation regression for the primary/safety dual-instance contract (buffer clock, CGM-dropout window, CGM-acceptance density)	SRS-ALG-001, SRS-ALG-008
TV-ALG-005	Unit (Algorithm)	Degraded/unavailable-input golden-vector replay	SRS-ALG-001, SRS-ALG-003
TV-ALG-006	Unit (Algorithm)	Meal/BG/intervention golden-vector replay	SRS-ALG-001, SRS-ALG-005
TV-ALG-007	Unit/Integration	Stateful continuity across persistence/reload/reset boundaries	SRS-ALG-004

Test ID	Level	Purpose	SRS Link
TV-ALG-008	Unit (Algorithm)	Boundary/sentinel cases (CGM, BG, pump) remain deterministic and safe	SRS-ALG-003, SRS-ALG-004
TV-ALG-009	Differential	Pregnancy config differential replay (target, upfront, TMAX) vs baseline	SRS-ALG-005
TV-ALG-010	Coverage	Structural coverage generation for Algo2015 and the bridge. Nominal targets are 100% bridge function/line/branch coverage and at least 98% Algo2015 function, 95% line, and 90% branch coverage; any shortfall requires a controlled reachability, safety-impact, mitigation, and disposition record in the STR exception package.	SRS-ALG-002
TV-ALG-011	Toolchain/Process	Static-analysis quality lane execution, and MISRA policy evidence closure as either linked report+deviations or explicit not-applicable decision rationale	SRS-ALG-006, SRS-ALG-007

Test ID	Level	Purpose	SRS Link
TV-ALG-012	Unit/Integration	A meal step with concurrent bolus/basal/meal output commands their exact sum, persists that total as the issued request, and accepts matching next-step pump feedback through Reconcile_I_Dose (I present, ~I absent); basal-only behavior remains unchanged	SRS-ALG-009
TV-OUTAGE-001	Unit	Outage BG-due scheduling: 120/60-minute interval selection at the >200/<80 thresholds, qualifying-glucose re-anchoring (CGM and fingerstick), blind-step/skip preservation, Codable persistence	SRS-OUTAGE-001, SRS-OUTAGE-005
TV-OUTAGE-002	Unit/Integration	Backup-basal bridge transitions: active mask cancel on connected forced-open, renewal suppression while released, pre-command re-arm on re-close, cancel-failure retry, re-arm-failure command block, full cancel-	SRS-OUTAGE-004, SRS-OUTAGE-006

Test ID	Level	Purpose	SRS Link
		hold-renew sequence order; bridged- exposure reconciliation itself (SRS- OUTAGE-006) is deliberately covered by the existing fallback- replay rows (TV- PUMP-008/TV- RUN-008 family) - the bridge introduces no new reconciliation evidence class, and release- window quantification at re-arm (engaged event updated in place with window duration + modeled schedule exposure, status resumed_without _reconciliation; rearm-failure keeps the window open - testRecloseQuantifiesTheReleasedWindowOnTheEngagedEvent) . Released- window pod- away gap reconciliation (SRS-OUTAGE- 006 v1.42, sponsor-directed 2026-07-18): reconnect after skipped steps builds the standard	

Test ID	Level	Purpose	SRS Link
		bounded schedule- weighted replay instead of a single chunk tuple - end-to- end scenario pins SimulatedDashP odOutageBridge ScenarioTests, including insulin conservation across gap lengths and rates; independent safety-review coverage in SimulatedDashP odOutageGapRep layReviewScena rioTests verifies persisted-credit conservation across force quit, exactly-once re- synthesis after a cadence-drift discard, quantized weighting for non-flat schedules, and engaged-event time-zone weighting; real- C++ chunk-vs- replay divergence measured and pinned in Algo2015Outage GapPodAwayRepl ayCharacteriza tionTests, whose engine- ledger conservation pins	

Test ID	Level	Purpose	SRS Link
		hold the C++'s booked row totals equal to the measured delta under the 0.01 U-quantized largest-remainder allocation; classifier row R-046 + testClassifierMatchesRuntimeForReleasedOutagePodAwayReconnectGapReplay keep the decision matrix and runtime in parity (plan-stamp authorization under the pending engaged event)	
TV-OUTAGE-003	Unit	Outage alert tiers and countdown surfaces: pre-scheduled due-soon/overdue monitoring (immediate upsert past deadlines, clearing on disable), preview-catalog copy pins, backup-basal informational supersession of the forced-open note, and the outage-active gate (offline-mode blind step true, relaunch-blip blind step false, fingerstick-	SRS-OUTAGE-002, SRS-OUTAGE-003

Test ID	Level	Purpose	SRS Link
		fed step true, mid-BG-run persistence, accepted CGM ends it, skip preserves state - BionicLoopRuntimeEngineOutageBGRunTests incl. testRelaunchBlindStepIsNotAnOutage)	
TV-PUMP-010	Unit	Pump-command watchdog: completion wins, timeout throws the supplied uncertain error, late completion after timeout is ignored safely, double completion ignored; pump-event delegate synchronous-completion contract pinned	SRS-PUMP-011
TV-CGM-001	Unit	Out-of-range CGM -> unavailable (-1) mapping	SRS-CGM-001
TV-CGM-002	Unit	Step-0 fresh/in-range gating	SRS-CGM-002
TV-CGM-003	Unit	Step>0 degraded run with unavailable CGM	SRS-CGM-003
TV-CGM-004	Unit/UI	Step-0 blocked-for-CGM path exposes explicit reason/state messaging to user surfaces	SRS-CGM-004, SRS-UI-002

Test ID	Level	Purpose	SRS Link
TV-CGM-005	Integration/System	Armed-loop absence of successful step execution beyond the approved interruption threshold is detected as a stalled-step condition using last-success/session-start timing	SRS-CGM-005
TV-CGM-006	Unit/Integration/UI	A stale bound G7 and a routine disconnect do not enter replacement mode; explicit replacement synchronously offers start/prior-ID/trigger evidence for persistence before returning to BLE acquisition; the remote-disconnect-during-authentication end-of-session heuristic enters persistent replacement mode; repeated triggers preserve the original episode clock; direct manager reconstruction restores replacement mode; authenticated adoption records old/new identity	SRS-CGM-006

Test ID	Level	Purpose	SRS Link
		and trigger, then clears replacement mode. Settings distinguishes routine BLE scanning from replacement acquisition, keeps Scan for new sensor visible and actionable throughout, and adds Scanning for new sensor... progress while replacement acquisition is active; retry restarts discovery without resetting the original evidence clock. The exact-freeze unit result remains unchanged. The September 3 adversarial review found that this method does not cover CGM-screen setup reentry, concurrent candidate ordering/single-winner behavior, automatic-trigger isolation, or first-reading algorithm use before staff comparison; those scenarios are not represented as	

Test ID	Level	Purpose	SRS Link
		passed. Under D06, they are covered by a controlled exercise required before first participant deployment.	
TV-BG-001	Unit/Integration	bgCheck creates/uses a single pending BG candidate without borrowing future slots beyond immediate next-step policy	SRS-BG-002
TV-BG-002	Unit/Integration	Submit after due-step execution rolls BG candidate to immediate next step and uses it there	SRS-BG-003
TV-BG-003	Unit/Integration	BG value maps to algorithm BGval while CGM mapping remains independent	SRS-BG-004
TV-BG-004	Unit/Integration	Pump unavailable during bgCheck blocks command application without overriding degraded policy	SRS-BG-005, SRS-PUMP-001
TV-BG-005	Unit/UI	Stale manual BG is rejected with explicit user-visible reason	SRS-BG-006
TV-BG-006	Unit/Integration	Telemetry records manualBG source, value, timestamps, and	SRS-BG-007, SRS-LOG-001

Test ID	Level	Purpose	SRS Link
		execution outcome	
TV-BG-007	Unit/Integration	Deferred from current software baseline. If step-0 BG rescue is enabled in a future accepted baseline, verify it executes only when policy gates pass.	SRS-BG-008
TV-BG-008	Unit/UI	Manual BG entry rejects values outside 20...600 mg/dL with explicit validation messaging, shows that manual BG is used for algorithm dosing decisions and does not calibrate CGM, and requires a spatially separate exact-value review confirmation whose Change action preserves the entered digits and whose entry action cannot submit under rapid repeated taps	SRS-BG-001
TV-BG-009	Unit/Integration	Pending BG candidate expires if not consumed on the immediate next target step	SRS-BG-009
TV-BG-010	Unit/Integration	New BG submission	SRS-BG-010

Test ID	Level	Purpose	SRS Link
		replaces existing pending candidate before execution	
TV-BG-011	Unit/Integration	Manual BG submit while loop is disarmed or masked-fallback reconciliation is pending does not dispatch runtime execution (bgCheck) or create pending BG state	SRS-BG-011
TV-BG-012	Unit/Integration	Manual BG submit before first successful anchored step is rejected and does not create pending BG state	SRS-BG-012
TV-BG-013	Unit/Integration/UI	Accepted persisted manual BG renders immediately as one unfilled marker at submission time, survives persistence/relaunch, becomes filled only from completed algorithm-input evidence, retains its original time without duplication, and omits incomplete, invalid, out-of-window, expired, replaced, or discarded unconsumed state	SRS-BG-002, SRS-BG-004, SRS-BG-009, SRS-BG-010, SRS-BG-013

Test ID	Level	Purpose	SRS Link
TV-CLIN-001	Unit/UI/Integration	While no usable subject configuration exists, initial clinical setup can be reviewed and saved without an unlock; that exception closes after first save. Subsequent Clinical Settings access/edits are gated by the offline clinical unlock verifier; material installation validates and stores verifier material, material refresh preserves accepted-counter state while clearing active unlock state, malformed/invalid/reused/non-ASCII codes block entry, accepted future counters unlock settings locally for the provisioned duration without requiring network access, and manual-lock storage failure does not falsely present a locked state	SRS-CLIN-001, SRS-CLIN-002, SRS-CLIN-013
TV-CLIN-002	UI/Smoke	Subject ID, Weight, Start Algo, and Same-Participant	SRS-CLIN-003

Test ID	Level	Purpose	SRS Link
		Reset (formerly Reset Algo) are presented in Clinical Settings and not in participant-facing settings sections	
TV-CLIN-003	Unit/UI	Target selector enforces allowed values (90, 100, 110, 120, 130 mg/dL) and rejects out-of-set values	SRS-CLIN-004
TV-CLIN-004	Unit/UI	Meal upfront selector enforces two-option set (75%, 90%) and maps selected value into runtime config	SRS-CLIN-005
TV-CLIN-005	Unit/UI	TMAX selector enforces 40...70 inclusive with 5-minute increments	SRS-CLIN-006
TV-CLIN-006	Unit/Integration	Clinical settings persistence restores values across relaunch with deterministic default/migration behavior	SRS-CLIN-007

Test ID	Level	Purpose	SRS Link
TV-CLIN-007	Unit/Integration	Start Algo and Same-Participant Reset (formerly Reset Algo) behavior remains unchanged after relocation into Clinical Settings	SRS-CLIN-008
TV-CLIN-008	Unit	Weight conversion and validation path stores kg from integer lbs UI input	SRS-VAL-001, SRS-CLIN-003
TV-CLIN-009	Unit/Integration	Clinical save-review semantics hold: no persisted/runtime config mutation before Save+OK, cancel keeps prior applied config, and persisted update appears in next step telemetry snapshot	SRS-CLIN-007, SRS-CLIN-008, SRS-LOG-001
TV-CLIN-010	Unit/UI	Participant-facing settings and the clinician target selector expose only the target set enabled by the clinician-selected target-access profile (Pregnancy vs Standard)	SRS-CLIN-009, SRS-CLIN-010
TV-CLIN-011	Unit/UI	Participant target change requires approval capture and blocks apply until approving	SRS-CLIN-011

Test ID	Level	Purpose	SRS Link
		staff name and approximate approval time are both recorded	
TV-CLIN-012	Unit/UI	Clinical Settings normalizes the draft target to an allowed profile value when the clinician changes the target-access profile	SRS-CLIN-012, SRS-CLIN-010
TV-CLIN-013	Unit/Integration	Target-access profile persists across save/relaunch/migration and is reflected consistently in both participant and clinician settings views	SRS-CLIN-007, SRS-CLIN-009, SRS-CLIN-010
TV-CLIN-014	Unit	First-launch defaults are Standard/120/75%/40 (clinical direction 2026-07-14) and normalization fallbacks track them; persisted participant values are never rewritten by defaults	SRS-CLIN-013
TV-CLIN-015	Unit/Integration	New Participant Reset clears every participant-scoped store (config, session keys, pump events, outbox, subject-scoped keychain unlock material) and returns to first-	SRS-CLIN-014

Test ID	Level	Purpose	SRS Link
		launch setup; refused while a pod is active (stores untouched); an armed session proceeds only through the guided stop-then-reset flow (explicit session stop, then the erase confirmation), with the engine-level stop-then-reset transition regression-pinned. Device-state isolation proves pending G7 replacement-acquisition persistence callbacks, alert work, and in-flight stall/adoption telemetry are invalidated before raw manager state is removed, so late completion cannot recreate outgoing-participant state (testNewParticipantResetInvalidatesAcquisitionPersistenceAndInFlightTelemetry). Telemetry drain-guard (SRS-CLIN-014 v1.41): undelivered/rejected outbox	

Test ID	Level	Purpose	SRS Link
		counts, drain-attempt-then-count summary, discarded-marker emission on the reserved-sequence path (offline-silent, skipped when nothing is pending), sequence-continuity-preserving spool erasure, discard-before-wipe ordering with no discard on blocked outcomes, and the pending-count consent line in the erase confirmation copy	
TV-CLIN-016	Unit	Weight plausibility bounds (SRS-CLIN-015): configurations with weight outside 50...500 lbs are not usable for arming (<code>testClinicalSettingsSavePolicyRejectsImplausibleWeightEntries</code>); the store top-clamps persisted weight at 500 lbs, preserves 0-as-unset, and never raises a lower entry (<code>testClinicalAlgorithmConfig</code>	SRS-CLIN-015

Test ID	Level	Purpose	SRS Link
		StoreTopClamps WeightAndPrese rvesUnsetAndSu bPlausible); the algorithm layer clamps consumed weight into 20...230 kg with telemetry visibility (testRealBUDos ingAlgorithmCl ampsImplausibl eWeightIntoSaf etyRailInTelem etry)	
TV- CLIN- 017	Unit/UI/Integr ation	Temporary Target verification shall cover explicit target/duration selection and allowed-value rejection; target- only, duration- only, and complete unactivated draft-loss detection; keep- editing and explicit discard navigation; app- standard selected-state presentation; placement above User and absence of unrelated participant-reset guidance; Pregnancy/armed /active- Pod/recovery activation gates; persistence, concurrent mutation	SRS-CLIN-016, SRS-ALERT-002, SRS-ALERT-006

Test ID	Level	Purpose	SRS Link
		serialization, deterministic replacement, early end, exact expiry, relaunch restoration, profile/session/reset clearing, New Participant Reset erasure, and historical execution-time lookup; primary-only target use with the safety target/fallback basis pinned to the permanent configuration; no session restart or immediate Pod command; notification schedule/cancel semantics; active Home/Settings state; and the specified physical-device Temporary Target and suspension/resumption scenarios. The combined-screen verification shall additionally prove that the existing suspend/resume command is directly available there, absent from normal Pod Settings, offers the unchanged four reminder choices without automatic resume, and	

Test ID	Level	Purpose	SRS Link
		disables action for unusable or transitioning Pod states. The suspend-ended alert shall expose the same resume command on Home and Alert Center, suppress duplicate in-flight commands, remain retryable after command failure, clear only through pump lifecycle confirmation, and escalate exactly once from actionable to safety-critical at 15 minutes with update telemetry and a fresh notification, including after relaunch; repeated source issue shall not restart or downgrade the escalation. Working automated symbols include testTemporaryTargetSelectionRequiresExplicitTargetAndDuration, testTemporaryTargetDraftWarningsWhenEitherSelectionWouldBeLost, testTemporaryTargetStorePers	

Test ID	Level	Purpose	SRS Link
		istsActiveOverrideWithoutMutatingClinicalConfig, testTemporaryTargetStoreSerializesConcurrentStateMutations, testTemporaryTargetStoreExpiresAtExactDeadlineAndRecordsDispositionOnce, testTemporaryTargetHistoryResolvesReplayRowsAcrossExpiry , testTemporaryTargetSessionStopClearsActiveOverrideButRetainsHistoricalExecutionContext, testCurrentLoopConfigUsesTemporaryTargetOnlyForPrimaryController, testTemporaryTargetActivationEligibilityRequiresPregnancyArmedAndActivePod, testTemporaryTargetRelaunchRestoresReminderAndForegroundExpiresAtDeadline, testBionicLoopPodSettingsHid	

Test ID	Level	Purpose	SRS Link
		eDuplicateInsulinSuspensionControl, testInsulinSuspensionPresentationUsesSuspendForActiveDelivery, testInsulinSuspensionPresentationUsesResumeForSuspendedDelivery, testInsulinSuspensionPresentationDisablesTransitionsAndUnavailablePod, testSuspendEndedAlertOffersDirectResumeAction, testResumeInsulinActionRunsOnCommandAtATimeAndRefreshesAfterSuccess, testResumeInsulinActionLeavesRetryAvailableAndSurfacesFailure, testSuspendEndedAlertEscalatesAfterFifteenMinutesAndRenotifiesOnce, testSuspendEndedAlertEscalatesImmediatelyAfterRelaunchPastDeadline, testRepeatedSuspendEndedIssueDoesNotDowngradeEscalatedAl	

Test ID	Level	Purpose	SRS Link
		ert, testUI003a_TemporaryTargetRequiresExplicitSelectionAndConfirmsActivation, testUI003b_ExerciseScreenOwnsInsulinSuspensionControl, testUI003c_TemporaryTargetWarnsBeforeDiscardingPartialSelection, and testUI006a_SuspendEndedAlertOffersResumeInulinAction. Suspend alert normalization is pinned by BionicLoopPumpAlertMapperTests; notification destination is pinned by BionicLoopAlertCenterRuntimeTests.testShowPreviewAlertsSupportsMultipleTypesAndPrecedence.	
TV-CLIN-018	Unit/UI	The explicit Standard-to-Pregnancy draft transition initializes meal upfront to 90%, Pregnancy-to-Standard and same-profile changes preserve the current valid	SRS-CLIN-017, SRS-CLIN-005

Test ID	Level	Purpose	SRS Link
		value, 75% remains selectable before save, cancel leaves persisted state unchanged, and an existing Pregnancy configuration is not migrated (testStandardToPregnancyTransitionDefaultsMealUpfrontToNinetyOnlyForThatTransition plus existing save/cancel persistence coverage).	
TV-PUMP-001	Unit	Pump unavailable -> run step, block command application	SRS-PUMP-001
TV-PUMP-002	Integration	Signal-loss policy persistence and clear behavior	SRS-PUMP-001, SRS-UI-002
TV-PUMP-003	Integration	Delivery reconciliation and min-dose quantization behavior	SRS-PUMP-003
TV-PUMP-004	System	Home pod card updates on connect/disconnect without entering settings	SRS-PUMP-004
TV-PUMP-005	Integration/System	Delivery-state clears from delivering via auto-refresh without opening Pump settings	SRS-PUMP-005
TV-PUMP-	UI/Integration	Closed-loop surfaces do not	SRS-PUMP-002

Test ID	Level	Purpose	SRS Link
006		expose manual bolus command paths	
TV-PUMP-007	Unit/Integration	Masked fallback maintenance blocks disruptive schedule writes during unsafe pump states, arms newly computed missing fallback before the same step's pump command only when refreshed pump status is known/available, arms persisted missing fallback before the next loop execution once a valid candidate exists, programs the secondary/safety q5 nominal-basal four six-hour fallback schedule when available, renews the 0 U/hr mask inside the 20-minute renewal window, permits the normal step command after a clean first-arm block with no schedule mutation, prevents same-cycle post-execution first-arm retry after a pre-command fallback event, blocks the normal	SRS-PUMP-006, SRS-PUMP-007, SRS-PUMP-008

Test ID	Level	Purpose	SRS Link
		step command when first-arm remask failure creates reconciliation-required recovery, treats post-schedule mask blocking/failure as recovery-required remask failure, defers maintenance when pump state or late bolus-in-progress command races block renewal but the current mask is still active, records offline fallback activation plus reconciliation-required recovery state when the mask later expires, and performs schedule refresh using reprogram-and-immediate-remask semantics	
TV-PUMP-008	Unit/Integration	Session reset/disarm attempts to restore the original programmed basal schedule, preserves recovery context when restore fails, retries connected restore/disarm	SRS-PUMP-009, SRS-STATE-004, SRS-STATE-005

Test ID	Level	Purpose	SRS Link
		when masked-fallback recovery remains pending on reconnect/start/foreground, preserves the existing session/cadence anchor after successful offline-expiry recovery while the loop remained armed, allocates credible same-pod pump-reported fallback delivery delta across missed primary/secondary algorithm replay steps with CGM=-1 and no catch-up pump commands, weights replay-step delivery by the persisted programmed fallback schedule including equal-rate, six-hour-window, partial-slot, and midnight-wrap cases while keeping the pump delta authoritative, suppresses replay for positive pump delta with no usable schedule weight, preserves zero-delivery replay rows when the authoritative	

Test ID	Level	Purpose	SRS Link
		pump delta is 0 U, suppresses replay for ambiguous, different/new pod, unknown identity, or history- discontinuous evidence while keeping same- session unreconciled continuation, records assumed- delivered modeled fallback exposure and queues bounded replay when the previous pod is known retired/inactive or past hard service-stop and pump-counter evidence is unrecoverable, stamps recovery- completion timing after restore finishes, captures basal- only reconnect evidence plus modeled-vs- pump-reported fallback delivery when compatible baseline data exist, and keeps explicit reset/loop-off recovery from silently turning the loop back on	

Test ID	Level	Purpose	SRS Link
TV-PUMP-009	Unit/Integration	Applied or uncertain automatic bolus commands persist issued-dose attribution; active delivery blocks live step advancement for the original/current pod; accepted no-boundary-cancel liveness coverage verifies DASH bolus completion/status convergence before the next 5-minute step for supported automatic bolus sizes and unavailable/unknown state on disconnect; matching same-request same-pod pump evidence replays missed primary/secondary algorithm steps before the live step with delivered insulin injected only at the first missed attribution step and CGM=-1; delivered-unit evidence outside 0...requested is non-credible and does not replay or advance; combined same-	SRS-PUMP-010, SRS-STATE-004, SRS-STATE-005

Test ID	Level	Purpose	SRS Link
		pod pump-total evidence is partitioned into pending issued-dose evidence plus fallback-basal residual only when the request, pod identity, completion timing, and residual fallback exposure are credible, while raw pump-total delivery remains telemetry/operator-review evidence; fallback replay overlap is monotonic and non-duplicating and uses the same delivered-unit credibility rule; absent, mismatched, non-credible, missing-identity outside the user-confirmed unavailable-pod clinical-policy path, or non-partitionable evidence skips live advancement without clearing the pending attribution; user-confirmed unavailable-pod meal evidence may be consumed without pod identity only	

Test ID	Level	Purpose	SRS Link
		when request step, units, delivered bounds, and non-active pump status match; and different/new pod evidence, including replacement-pod active-delivery status, records different_or_new_pod, preserves the current algorithm session, assumes the issued dose was delivered, feeds the assumed delivered amount into the first eligible attribution step by replay or live input, scrubs unrelated replacement-pod lastDelivery from the resumed live algorithm input, clears the old-pod pending attribution after consumption, and allows later insulin-adding automatic commands including automatic resume under the new clinical-policy disposition; fresh-completion upgrade rule (SRS v1.48, field	

Test ID	Level	Purpose	SRS Link
		incident 2026-07-20): the post-relaunch persisted-evidence deferral/suppression is superseded by a fresh same-pod settled status at/after expected physical completion with bolusNotDelivered = 0, surfacing the pump-observed completed delivery so a frozen capped-in-progress or assumed figure cannot under-report the meal evidence, while canceled partials (nonzero register), pre-completion early-reads, active delivery, and different/unproven pods keep the deferral, and matching non-assumed sub-requested evidence is replaced by the strictly-higher same-pod post-completion observation before consumption - pinned end-to-end with the incident telemetry in	

Test ID	Level	Purpose	SRS Link
		BionicLoopPump EvidenceRelaunchUpgradeTests	
TV-MEAL-001	Unit	Meal announce borrow-window gating after cadence reconciliation against algorithm-reported next step	SRS-MEAL-001, SRS-RUN-002
TV-MEAL-002	Unit/Integration/Characterization	Meal announce is blocked when pump is delivering, suspended, or unknown; when the primary algorithm is already forced-open without an exact-step pending BG; or when the selected borrowed/due step would be the characterized 13th glucose-blind step. A connected active .delivering status maps to pump-busy, .suspended maps to resume-before-meal guidance, and an idle unresolved attribution remains blocked with reconciliation guidance. Suspended-path coordinator	SRS-MEAL-002, SRS-BG-004

Test ID	Level	Purpose	SRS Link
		coverage proves rejection occurs before meal persistence, either algorithm, cadence advancement, or pump command, and subsequent live input contains no rejected meal request/delivery; a fresh idle refresh after Resume makes availability immediate without a loop step. The 12th blind step remains eligible; fresh CGM or a valid 20 . . . 600 mg/dL BG targeting the prospective step permits execution. Combined BG + meal still feeds both primary and safety algorithms exactly once and clears forced-open in the real C++; stale/future/invalid BG and pump signal loss remain blocked.	
TV-MEAL-003	Unit/Integration	Meal announce executes on current due step when slot is already due/missed	SRS-MEAL-004

Test ID	Level	Purpose	SRS Link
TV-MEAL-004	Unit	Meal announce rejected before first successful anchored step	SRS-MEAL-005
TV-MEAL-005	Unit	Meal unavailable reason precedence reports noPump before signalLoss when no active pod is present	SRS-MEAL-002, SRS-UI-002
TV-MEAL-006	Unit/UI	Meal unavailable messaging includes explicit actionable reason and retry timing when blocked, including reconnect/recovery-required guidance while masked-fallback reconciliation is pending	SRS-MEAL-003, SRS-UI-002
TV-MEAL-007	Unit/UI	Meal composer revalidates availability on foreground refresh, qualifying-glucose publication, live pump-status change, and immediately before submit so stale available or blocked state cannot dispatch an invalid meal request or retain incorrect guidance; open-composer revalidation re-	SRS-MEAL-006, SRS-UI-002

Test ID	Level	Purpose	SRS Link
		arms in place or renders inline blocked messaging, and when a loop step consumes the pending fingerstick BG that unlocked the composer it holds the armed composer on a passive waiting notice until the exact step's qualifying-glucose marker publishes (including an intermediate checkpoint that already reports the step executed), never demanding another fingerstick within the BG validity window; connected active delivery promptly presents pump-busy guidance	
TV-MEAL-008	Unit/UI/Integration	Meal submit does not present success until runtime result is known; blocked/rejected and uncertain outcomes surface explicit user-facing recovery messaging	SRS-MEAL-007, SRS-UI-002

Test ID	Level	Purpose	SRS Link
TV-MEAL-009	Integration	Pending or uncertain meal request state, including correlated flow ID, persists across relaunch and prevents duplicate meal entry until resolved	SRS-MEAL-008, SRS-MEAL-009, SRS-STATE-001
TV-MEAL-010	Integration/System	Command-outcome uncertainty (timeout/error with unresolved delivery state) blocks repeat meal announce and surfaces explicit operator guidance until reconciliation; immediate-success and reconciled meal lifecycle closure remain replayable across terminate/relaunch windows until resolved telemetry is emitted	SRS-MEAL-008, SRS-MEAL-009, SRS-PUMP-001
TV-MEAL-011	Integration/System	Competing-trigger slot conflict does not silently lose or reinterpret meal intent; user receives explicit slot-conflict blocked/retry feedback and no hidden reassignment to a	SRS-MEAL-010, SRS-RUN-002, SRS-UI-002

Test ID	Level	Purpose	SRS Link
		different borrowed step	
TV-MEAL-012	Unit/UI/Integration	When meal entry is opened during active bolus delivery, the app presents a destructive Home inline cancel-delivery flow, keeps that flow visible while active meal delivery remains in progress, reports actual delivered insulin after cancellation in the Home summary region, retains that summary until both the next later algorithm step and a 5-minute minimum display window have passed, renders active in-progress meal delivery in the normal meal-dose color while reserving caution color for actual interrupted delivery, and preserves the delivered amount for subsequent algorithm-step accounting	SRS-MEAL-011, SRS-PUMP-003, SRS-UI-002
TV-MEAL-013	Unit/UI/Integration	When meal delivery progress cannot be confirmed for the original pod, the app offers an	SRS-MEAL-012, SRS-PUMP-010, SRS-UI-002

Test ID	Level	Purpose	SRS Link
		explicit pod-replacement escape only after known old-pod unavailability or expected-completion timeout, suppresses that escape for non-meal issued-dose attributions and already matched evidence, records assumed-delivered evidence with user_abandoned_unavailable_pod, clears only meal-progress UI state, preserves issued-dose attribution for replay/live-step accounting, and routes the operator to pod setup	
TV-STATE-001	Integration	Relaunch restores cadence and algorithm state	SRS-STATE-001
TV-STATE-002	Integration	Reset clears all session state and starts fresh	SRS-STATE-002
TV-STATE-003	Integration/System	Pump and CGM manager state persistence supports reconnect without forced re-pairing on relaunch	SRS-STATE-003
TV-STATE-	Integration	Masked fallback persistence	SRS-STATE-004

Test ID	Level	Purpose	SRS Link
004		retains original/fallback schedules, maintenance timestamps, active safety target, connected pod identity and total-delivery baseline, restore-failed or reconciliation-required recovery context, primary and secondary/safety q5 nominal-basal profiles, and any pending pump-delta reconciliation state needed for later refresh, restore, reconnect recovery retry, confirmed/corrected missed-step algorithm replay, or ambiguous unreconciled resume	
TV-STATE-005	Unit/Integration	Runtime recovery, pump reconnect, pod replacement, CGM recovery, fallback reconciliation failure, launch, foreground, scheduler wake, telemetry replay, and cloud/auth state do not stop/start/reset the algorithm or create a	SRS-STATE-005

Test ID	Level	Purpose	SRS Link
		replacement algorithm session without explicit operator action; unreconciled fallback recovery preserves the existing session/cadence state while suppressing replay	
TV-LOG-001	Unit	Step telemetry contains explicit <code>step_executed_at</code> plus input/output/command fields; Recent Dose Steps identifies a valid persisted manual BG as <code>Fingerstick</code> , shows the exact value used, preserves concurrent CGM, and shows no fingerstick marker when the BG sentinel is absent. The current-session SQLite archive retains and exports every step beyond the 576-row presentation cache, restores the newest cache rows after relaunch, supports historical-row update, reads the full archive for CSV generation	SRS-LOG-001

Test ID	Level	Purpose	SRS Link
		off the main actor, and clears on explicit algorithm-session reset; byte-marker and queue-affinity regressions verify that archive database artifacts no longer contain the cleared participant marker and full-history export cannot block the main actor (LoopTelemetryStoreTests.testClinicalStepArchiveKeepsFullHistoryWhileUICacheAndRelaunchStayBounded, testClinicalStepArchiveCSVReadRunsOffMainActor, testAlgorithmSessionResetClearsFullClinicalStepArchive, testAlgorithmSessionResetRemovesArchivedClinicalBytesFromDatabaseArtifacts).	
TV-LOG-002	Integration	CSV export schema and row append behavior, including masked-fallback missed-step algorithm replay	SRS-LOG-002

Test ID	Level	Purpose	SRS Link
		rows from credible pump delta with CGM=-1, per-step delivered-insulin input including explicit 0 u input for replay steps with no pump allocation, and no catch-up pump command rows	
TV-LOG-003	Unit/Integration	Async export avoids main-actor blocking	SRS-LOG-003
TV-LOG-004	Unit	Cloud-log upload filter is inclusive (selected level and higher severities) with default Error; no in-app logging control exists (UI removed 2026-07-16); effective-threshold precedence session > remote override > default is pinned	SRS-LOG-004
TV-LOG-011	Unit	Remote logging config as the sole lever: response parsing (apply valid including the dashboard's fractional-milliseconds toISOString expiry shape, clear on null/expired server config, ignore malformed or unknown level), persistence	SRS-LOG-011

Test ID	Level	Purpose	SRS Link
		apply/clear/ignore -preserves, override activation via the effective- threshold policy with legacy session storage ignored, upload- logger threshold honoring the server config, change-observer notification (fires on apply/clear change, silent on unchanged re- fetch), periodic re-fetch loop (re- fetches on interval, applies the fetched config, stops on cancel), and the read-only Settings status text (server config > default precedence with source attribution; legacy session bytes never shown; phone- set non-default level surfaced) - BionicLoopClou dLogPolicyInfr astructureTest s incl. testCloudLogUp loadPolicyIgno resLegacyInteg rationSessionS torage, testCloudLogUp loadLoggerHono rsRemoteOverri	

Test ID	Level	Purpose	SRS Link
		<code>deLevel,</code> <code>testRemoteConfigRefreshNotifiesObserverOnlyWhenStoredOverrideChanges,</code> <code>testStartPeriodicRefreshKeepsRefetchingUntilCancelled,</code> <code>testStatusDescriptionShowsRemoteConfigThenDefault,</code> <code>testStatusDescriptionShowsPhoneSetLevelWhenNoOverrides</code>	
TV-LOG-005	Unit	Clinical Settings save flow emits deterministic <code>ui.critical</code> telemetry (<code>state_viewed</code> , <code>submit</code> , <code>cancel</code> , <code>blocked</code>) with stable element IDs and old/new value details	SRS-LOG-005
TV-LOG-006	Unit/Integration	App lifecycle telemetry includes timezone and clock-check context fields with correct trigger semantics (<code>launch</code> , <code>foreground</code> >24h gate, <code>timezone_or_time_changed</code>)	SRS-LOG-006
TV-LOG-007	Unit/Integration	Meal-request telemetry exposes the	SRS-LOG-007, SRS-MEAL-007

Test ID	Level	Purpose	SRS Link
		implemented lifecycle transitions (submitted, accepted, success, blocked, uncertain, resolved) without optimistic-success duplication, with replay durability across terminate/relaunch windows, and loop-command telemetry preserves explicit command outcome semantics (applied, blocked, uncertain)	
TV-LOG-008	Unit/UI	Target-access-profile and participant approval-capture telemetry emit stable <code>ui.critical</code> events with required detail fields (<code>target_range_profile</code> , requested/applied target, approval metadata, and blocked/cancelled reason)	SRS-LOG-008
TV-LOG-009	Unit/UI/Integration	Fallback review telemetry persists and renders fallback	SRS-LOG-009

Test ID	Level	Purpose	SRS Link
		arm, maintenance-deferred, renew, schedule-unchanged check, refresh, mask-expiry, disarm, and restore/remask failure events with rate/source/target/duration/reconciliation/pod-continuity detail in Recent Dose Steps; resolved recovery rows include modeled expected delivery, same-pod pump-reported delivered insulin when available, and pump-delta reconciliation detail when confirmed/corrected recovery occurs; executed step rows render both primary and secondary algorithm summaries (step count, suggested dose, nominal basal, instant basal) from persisted per-step telemetry; confirmed/corrected reconnect recovery records no-command missed-step primary/secondary algorithm	

Test ID	Level	Purpose	SRS Link
		replay rows with CGM=-1 and persisted- schedule- weighted per- step pump delivery including explicit 0 u replay steps when no pump allocation belongs to a missed step, and the resumed live step uses actual refreshed pump status without duplicate recovered- delivery injection; and cloud telemetry emits a structured loop.fallback. event family with stable fallback_event _id, delivery/reconcili ation summary fields including pod_continuity , programmed schedule entries, safety q5 profile metadata including per-six- hour-segment observed-slot counts on arm/refresh/unch anged events, the lean unchanged- check payload (schedule + coverage without the bulk 288-rate	

Test ID	Level	Purpose	SRS Link
		array) at the steady 6-hour cadence, Recent Dose Steps schedule-row outcome labels and per-segment observed/imputed provenance, and duplicate suppression for unchanged fallback review state	
TV-LOG-010	Unit	Rescinded 2026-07-16 with the SRS-LOG-010 supersession (build-baked profile and on-phone session layer removed pre-ship; the covering tests were deleted with the code). Row retained for ID stability; legacy-storage-ignored behavior is covered under TV-LOG-011.	SRS-LOG-010
TV-LOG-012	Unit/Integration/Load	Sequence-numbered envelopes and rejection/recovery evidence (SRS-LOG-012): envelope bytes and durable sequence match; SQLite migration preserves prior rows/counters/restore state, purges the source only after commit, and	SRS-LOG-012

Test ID	Level	Purpose	SRS Link
		imports a later fallback spool idempotently; retained clinical rows are never count-evicted while chatter rotates; interrupted inflight rows normalize after relaunch; durable emit returns before suspended network drain; immediate work does not cancel an older retry; and 10,000 retained events survive relaunch and drain exactly once in 200 bounded batches. Batch tests pin unchanged embedded envelope bytes, 50-event/512-KiB bounds, duplicate acknowledgements, mixed retryable/permanent outcomes, subject-conflict routing, large-event single upload, and HTTP 404/405/501 fallback to the unchanged single-event route. Existing coverage continues to pin	

Test ID	Level	Purpose	SRS Link
		retained-first recovery, aggregate chatter-drop evidence, passive Settings status, notification deduplication, file protection/backu p exclusion, reset erasure including absence of a prior-participant byte marker from the recreated database/WAL/sh ared-memory artifacts, required envelope identity, structured permanent 4xx evidence, retryable 429, isolated Subject ID Conflict, legacy decode, and retired-alert migration. Working symbols are in BionicLoopTele metryOutboxRec overyTests, BionicLoopClou dTelemetryInfr astructureTest s, and BionicLoopTele metrySpoolRete ntionTests; the Scout producer fixtures come from the real batch handler. The local transport contract is verified; Scout	

Test ID	Level	Purpose	SRS Link
		observations are supportive and are not a formal closure condition.	
TV-LOG-013	Unit/Integration/System	Temporary Target telemetry verification shall cover durable exactly-once lifecycle enqueue with subject override, first-use application evidence, per-step permanent/applied/safety target fields, execution-time identity on replay rows, reset drain/erasure behavior, real app payload fixture ingestion, BionicScout active/awaiting/in-use/inactive projection, and delayed/out-of-order terminal-state protection. Working app symbols include <pre>testTemporaryTargetLifecycleTelemetryEmitsDurableContextAndClearsAfterEnqueue, testLoopStepTelemetryIdentifiesTemporaryTargetUseAndPermanentSafetyBasis, testReplayTelemetryDoesNotLa</pre>	SRS-LOG-013, SRS-CLIN-016

Test ID	Level	Purpose	SRS Link
		belPermanentBasisSafetyInputAsTemporaryTargetUse, testReplayTelemetryRetainsExpiredTemporaryTargetIdentityByExecutionTime, and TemporaryTarget cloud-envelope infrastructure tests. Captured app payload fixtures support the local contract checks; Scout projection observations are corroborative and are not a formal closure condition.	
TV-LOG-014	Unit/Integration	Acquisition-stall telemetry uses stable reason/threshold/start/trigger/prior-sensor fields and successful-adoption telemetry uses stable old/new/trigger/time fields. Both use stable event IDs through <code>cgm.state.changed</code>, suppress concurrent in-flight duplicates, persist a marker only after durable enqueue, survive relaunch without reserving a second wire	SRS-LOG-014

Test ID	Level	Purpose	SRS Link
		sequence, permit retry after failed enqueue, and allow a later acquisition/adopti on event. Scout observation is corroborative and is not a formal closure condition.	
TV-LOG-015	Unit/Integrati on/UI	LegacyAlgorith mArtifactExpor tTests shall verify documented-artifact discovery/groupin g, matrix epoch association, historical selection integrity, export-time re-discovery, byte-exact stored ZIP entries and CRC, bounded live append/growth behavior, archive naming, and empty algorithm-set handling. RecentDoseReco veryArchiveTes ts shall verify that the complete CSV and only the exact requested algorithm epoch enter one ZIP, an unavailable explicit epoch produces CSV-only output, absent inspection state selects only the newest valid grouped epoch,	SRS-LOG-015, SRS-SEC-001

Test ID	Level	Purpose	SRS Link
		bytes appended to the CSV during copy remain outside the snapshot, and a missing CSV produces no archive. App integration shall verify the existing Recent Dose Steps <code>ShareLink</code> remains DEBUG-visible and otherwise clinical-unlock-gated, prepares the archive off the main actor without another modal/sheet/activity-controller wrapper, and removes temporary staging on both completed-view dismissal and canceled preparation. No separate physical share/open claim is retained in the current submission package.	
TV-UI-001	UI/System	Home loop-state precedence rendering and cadence-phase age classification (<code>nextDueAt</code> -based <code>Active/Aging/State</code>)	SRS-UI-001
TV-UI-002	UI/System	Availability messaging	SRS-UI-002

Test ID	Level	Purpose	SRS Link
		matches runtime outcomes	
TV-UI-003	UI/System	CGM/Pod setup modal Cancel dismisses directly and does not force settings on no-active-pod startup	SRS-UI-003
TV-UI-004	Unit/UI	Meal announcement composer auto-cancels on app background transition	SRS-UI-004
TV-UI-005	UI/Smoke	Home primary controls are present and actionable in deterministic launch mode (settings, manual BG, Let's Eat); Manual BG and Meal Announcement remain visible, hittable, and screen-position-stable while expanded alert/status/chart content scrolls	SRS-UI-002
TV-UI-006	UI/Smoke	Home settings and manual-BG sheets can be opened and dismissed without dead-end navigation	SRS-UI-002, SRS-BG-001
TV-UI-007	Unit/UI	CGM display masks stale (>11m) or unreliable	SRS-UI-005

Test ID	Level	Purpose	SRS Link
		(hasReliableGlucose == false) readings as -- and hides trend arrow	
TV-UI-008	Unit/Integration	UTC clock-drift warning behavior: >600s skew emits non-blocking actionable warning with 24h rate limit, <=600s shows no warning, and unavailable checks do not spam warnings	SRS-UI-006
TV-UI-009	Unit/UI	CGM value formatting maps boundaries to textual LOW/HIGH across display surfaces and suppresses unit suffix for those states	SRS-UI-007
TV-UI-010	Unit/UI	Home CGM chart uses bounded dynamic y-axis maxima (300/350/400) based on displayed peak values and a point-only trace presentation with connector stroke and connected area fill disabled	SRS-UI-008
TV-UI-011	Unit	Server-configurable investigational badge: parse pinned against the captured	SRS-UI-009

Test ID	Level	Purpose	SRS Link
		deployed-lambda response (apply text, apply explicit blank, clear on null, ignore missing-text/malformed), three-state effectiveText (absent -> built-in default, text verbatim, blank -> nil/no capsule, clear -> default), refresh apply/clear/failure -inert, and the shared periodic loop fetching both server configs - BionicLoopCloudBannerConfigTests + testStartPeriodicRefreshKeepsRefetchingUntilCancelled; Scout-side route coverage test_banner_config.py (7) and dashboard badge/panel coverage monitoringChromeBadges.test.tsx / bannerConfigApi.test.ts (BionicScout repo)	
TV-ALERT-001	Unit	Alert normalization maps Omni/G7/runtime events to	SRS-ALERT-001, SRS-ALERT-002

Test ID	Level	Purpose	SRS Link
		canonical model fields	
TV-ALERT-002	Unit/Integration	Alert precedence keeps critical alerts visible when lower-severity alerts coexist and places active pump signal loss above a newer same-severity check-BG prompt while safety-critical severity remains dominant	SRS-ALERT-003, SRS-ALERT-011
TV-ALERT-003	Integration	Transient reconnect events are debounced/coalesced without suppressing persistent faults	SRS-ALERT-004
TV-ALERT-004	Integration/System	Alert clear/ack rules behave per alert type and update UI state correctly	SRS-ALERT-005
TV-ALERT-005	System/Manual	Protocol-required alerts and wording are present and actionable in app flows	SRS-ALERT-006
TV-ALERT-006	Unit/Integration	High-priority non-CGM alerts emit background local notifications with dedupe/cooldown , while CGM alerts and informational alerts do not	SRS-ALERT-007

Test ID	Level	Purpose	SRS Link
TV-ALERT-007	Unit/UI	Alert Center shows active and recently-cleared alerts with deterministic sorting and acknowledge path for required-ack alerts	SRS-ALERT-008, SRS-ALERT-005
TV-ALERT-008	Integration	Pump/CGM persisted-alert lifecycle hooks preserve issued/unretracted/retracted state across relaunch and restore active alert visibility	SRS-ALERT-009
TV-ALERT-009	Unit/Integration	Time-sensitive alert state refreshes at minute cadence while active: countdown wording updates without notification spam, and persisted suspend-ended state escalates once at its 15-minute deadline with a fresh safety-critical notification	SRS-ALERT-010, SRS-ALERT-007, SRS-CLIN-016
TV-ALERT-010	Unit/UI	Home alert-stack presentation preserves severity, condition-rank, then recency ordering (including pump signal loss above same-severity check-BG),	SRS-ALERT-003, SRS-ALERT-011

Test ID	Level	Purpose	SRS Link
		renders the top alert with compressed peek and explicit expansion, applies the root-cause suppression matrix (Home surface only, never safety-critical), and surfaces the hidden related-alert count line whenever suppression hides any active alert	
TV-ALERT-011	Unit/Integration	No-active-pod cleanup retracts only non-critical pod-tied alerts while retaining ALERT-PUMP-FAULT and ALERT-PUMP-INCOMPATIBLE until explicit closure	SRS-ALERT-005, SRS-ALERT-012
TV-ALERT-012	Integration/System	Algorithm Stepping Interrupted issues an actionable alert, clears on resumed successful stepping or loop disarm, and remains distinct from informational G7 unavailable/failed status surfaces	SRS-ALERT-013, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005

Test ID	Level	Purpose	SRS Link
TV- ALERT- 013	Unit/Integration/System	Algorithm Stepping Interrupted issues after >15 minutes without successful step execution while armed, carries blocker/root-cause detail from the latest unresolved runtime outcome/current authoritative pump condition, and clears on next successful step or loop disarm. Verification shall cover stale persisted CGM/pump blocker restoration, fallback-recovery replacement of stale CGM detail even with a broad CGM lifecycle alert active, suspension precedence while the suspend-ended alert remains active, fallback-specific copy after resume, relaunch persistence without a newer result, one-dedupe-key content transition, successful-publication	SRS-ALERT-014, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-UI-002

Test ID	Level	Purpose	SRS Link
		blocker/alert clearing, and Home suppression of the redundant interruption banner while retaining the stronger suspension alert. Pump-status recovery coverage shall additionally prove that a newer active-Pod idle refresh after a <code>pumpStatusUnavailable</code> attempt updates the same alert to communication-restored/waiting-for-glucose copy without triggering <code>doWork</code> ; repeated newer idle polls leave the alert timestamp unchanged; stale, unknown, delivering, suspended, and no-active-Pod states do not claim recovery; recovered-waiting state persists across relaunch; a later unknown status restores unavailable guidance; and the next successful step clears both blocker and alert.	

Test ID	Level	Purpose	SRS Link
TV-ALERT-014	Unit/Integration	CGM availability/failure normalized alerts remain informational in-app status only, do not expose required-ack behavior, and never schedule background local notifications	SRS-ALERT-015
TV-ALERT-015	Unit/Integration	App-derived CGM urgent-low review alert issues only for trustworthy G7 readings <55 mg/dL, preserves reviewed state while active, auto-clears on trustworthy recovery >=55 mg/dL, persists acknowledged active state across reset/reattach, and never schedules background local notifications	SRS-ALERT-016, SRS-ALERT-005, SRS-ALERT-007
TV-ALERT-016	Unit/Integration	No-active-pod conditions raise a safety-critical pump alert after debounce, repeat background local notification attempts at the approved 30 minute cadence while the condition remains true, and stop repeating when	SRS-ALERT-017, SRS-ALERT-007, SRS-ALERT-005

Test ID	Level	Purpose	SRS Link
		active pod state is restored	
TV-ALERT-017	Unit/Integration/UI	Algo2015 checkBG output surfaces a deduped prompt stating that either a fingerstick BG check or sensor read is needed for automated dosing to resume, with foreground haptic feedback and Home BG-button emphasis but no inline Enter BG action; it clears on BG entry or flag clear, suppresses repeat alerts while active, does not restore stale active state after relaunch, and preserves unrelated alert actions. The shared emphasis policy includes check-BG/duesoon/overdue direct requests and excludes generic backup-basal/pump/sens or alerts. The accepted breathing state changes only the BG control's centered caution-amber shadow bloom while its ordinary card shadow remains	SRS-ALERT-018, SRS-ALERT-005, SRS-ALERT-007, SRS-UI-002

Test ID	Level	Purpose	SRS Link
		static; Reduce Motion disables the bloom and retains the static border/wash (testAlgorithmCheckBGRequestSurfacesDedupe dHapticAlertAndClears, testHomeAlertBannerPrimaryActionPolicyOmitsCheckBGButtonAndRetainsLogin, testManualBGRequestEmphasisCoversOnlyDirectFingerstickRequests, testManualBGRequestPulseAnimatesOnlyLocalizedHalo)	

Controlled clarification for TV-PUMP-009 (v1.59): the fresh-completion regression set includes `testCanceledPartialBolusEvidenceSurvivesZeroNotDeliveredRegisterAfterRelaunch`, `testExplicitInProgressCapAllowsPostCompletionUpgrade`, `testEngineKeepsCanceledPartialEvidenceUnchanged`, `testEngineKeepsAssumedEvidenceUnchanged`, `testIssuedDoseEvidencePersistsCanceledObservationContext`, `testIssuedDoseEvidenceDecodesLegacyRecordWithoutObservationContext`, and `testIssuedDoseEvidencePersistsCanceledObservationContext`. Together they pin that a zero bolusNotDelivered register is not completion proof after cancellation; only explicitly persisted in-progress-capped provenance may authorize a higher settled observation, while canceled and legacy-unclassified partial evidence remains authoritative.

Controlled clarification for TV-PUMP-009 (v1.71):

`LoopRuntimeCoordinatorPumpCommandClassificationTests.testDoWorkMapsBlockedPumpCommandExecutionErrorToBlockedOutcome` reproduces a locally started request that is definitively rejected and otherwise appears as a fabricated full delivery; it verifies matching cleanup and no surviving `lastDelivery`. The uncertain-outcome companion verifies no cleanup. `BionicLoopPumpServiceAdapterDisconnectTests.testDefinitivelyBlockedBolusCle`

arsOnlyMatchingCachedRequest pins request-step/unit matching so cleanup cannot erase a newer request. | TV-ALERT-018 | Unit/Integration/Characterization | Algo2015 forced-open-loop output is reachable under sustained missing-CGM characterization, blocks automated pump commands after pre-command fallback maintenance, records algorithm telemetry, surfaces a safety-critical recovery alert, persists that alert across skipped/no-telemetry publications, and clears only on explicit later inactive forced-open output | SRS-ALERT-019, SRS-ALERT-003, SRS-ALERT-005, SRS-ALERT-007 || TV-ALERT-019 | Unit/Integration | G7 replacement acquisition has no alert before ten minutes, raises the approved actionable copy at the exact boundary, routes alert navigation to CGM Settings, does not restore stale LoopKit alert payloads, and auto-clears when replacement mode ends. The delayed evaluation task is canceled on manager detach/deletion and the condition does not generate a BionicLoop background CGM notification. | SRS-ALERT-020, SRS-ALERT-015 || TV-SEC-001 | Unit/Integration | Local export controls: file-sharing keys absent from the built bundle (testAppInfoPlistDoesNotEnableDocumentsFileSharing), CSV export in Application Support with complete-protection writing options + backup exclusion + legacy Documents cleanup (LoopTelemetryStoreTests), spool until-first-unlock protection + backup exclusion (testFileStoreSpoolCarriesUntilFirstUnlockProtectionAndBackupExclusion), and protection/backup-exclusion application to enumerated Algo2015 diagnostic artifacts (testAlgo2015TailReadAppliesArtifactProtectionToDiscoveredFiles) | SRS-SEC-001, SRS-SEC-002 | Controlled Build 843 clarification (v1.91): the dependency-control companion checks CryptoSwift's official source URL, exact 1.10.0 project requirement, resolved revision, and tracked lockfile status. Build 843 is the IDE submission build designated on 2026-09-03. Build 843 evidence and traceability were accepted under D08 on 2026-09-07 EDT. | TV-SEC-002 | Integration/System | Deferred from current software package. If Scout becomes relied upon for required study data or safety decisions, verify the defined phone-to-cloud telemetry contract, upload control behavior, completeness, and failure handling. | SRS-SEC-001 || TV-SEC-003 | Integration/System | Deferred from current software package. If protected cloud API access is re-entered into scope, verify it requires valid authenticated session. | SRS-SEC-003, SRS-SEC-006 | | TV-SEC-004 | UI/Integration | Deferred from current software package. If multi-provider onboarding is re-entered into scope, verify allowed sign-in entry points and failure states. | SRS-SEC-004, SRS-SEC-006 || TV-SEC-005 | Integration/System | Deferred from current software package. If authorization-role enforcement is re-entered into scope, verify unauthorized telemetry/dashboard actions are denied. | SRS-SEC-005, SRS-SEC-006 || TV-SEC-006 | Unit/Integration | Deferred from current software package. If password-recovery workflow is re-entered into scope, verify reset-code request and confirm-reset success/failure handling. | SRS-SEC-007, SRS-SEC-006 || TV-SEC-007 | Unit/Integration | Deferred from current software package. If launch session restore is re-entered into scope, verify authenticated UX is preserved when token recovery succeeds. | SRS-SEC-008, SRS-SEC-006 || TV-SEC-008 | Unit/UI | Deferred from current software package. If auth-failure Home-bypass continuity is re-entered into scope, verify signed-out active therapy keeps persistent top-level Home login access independent of

alert order; Home/Alert Center/Account & Session and notification routes enter login; and recovery presentation does not stop/reset the local algorithm session. | SRS-SEC-009, SRS-SEC-006 |

5.0 Algo2015 Structural-Coverage Campaign

- `Algo2015DualInstanceIsolationTests`
(`testInterleavedSafetyStepsDoNotAdvancePrimaryBufferPosition`,
`testPrimaryDropoutClockRunsOnWallStepsWithSafetyTrackInterleaved`,
`testPrimaryCGMAcceptanceTrackingStaysDenseWithSafetyTrackInterleaved`) pins the two-live-instance isolation contract against the real C++ pairing: the primary's buffer position advances exactly one per primary step with the safety track interleaved, the CGM-dropout tolerance forces open loop exactly on the 13th blind step (65 min > the 60-minute `Dropout_LimitT`), and CGM-acceptance tracking stays dense (TV-ALG-004 supporting evidence and shared-state regression guard).
- `Algo2015ManualBGDropoutClockCharacterizationTests`
(`testManualBGAAfterForcedOpenClearsAndRestartsTheFullWindow`,
`testManualBGMidDropoutRestartsTheWindowFromTheBGStep`) characterizes the fingerstick-BG dropout-clock contract against the real C++: an accepted BG clears forced-open on its own step, catches the acceptance index up to the buffer position, and restarts the full 60-minute window (12 further blind steps tolerated; forced-open on the 13th after the BG). This supports SRS-OUTAGE-001..006, SDD-OUTAGE-001, and TV-ALG-004.
- Fingerstick-supported CGM-outage working coverage (2026-07-14):
`OutageBGRunPolicyTests` (11), `MaskedFallbackMaintenancePolicyTests` released-for-outage gates (2), `BionicLoopRuntimeEnginePrePumpCommandMaintenanceTests` bridge transitions + full sequence (5), `BionicLoopRuntimeEngineOutageBGRunTests` context/outage-detection/alert tiers (11), `BionicLoopAppAlertDomainSupportTests` outage preview pins, `PumpCommandWatchdogTests` (5),
`BionicLoopPumpDelegateContractTests` (2).
- Build-profile working coverage (2026-07-15) - **retired 2026-07-16 with the SRS-LOG-010 supersession**: the parse and launch-reconcile pins were deleted with the profile/session code (the former cloud-log session infrastructure test class was removed; its threshold-filter coverage moved into `BionicLoopCloudLogPolicyInfrastructureTests`). The Scout-side `test_CT_J5_013...` ingest pin remains valid for `app_version/build_number` on the CloudWatch line and tolerates the now-unused session metadata keys.
- Remote-config + diagnostics working coverage (2026-07-16):
`BionicLoopCloudLogPolicyInfrastructureTests` remote-config parsing/apply pins (2),
`BionicLoopUserNotificationAppAlertSchedulerTests` authorization-status enrichment

pins (2), PumpCommandWatchdogTests timeout-observer pins (2); Scout-side route coverage test_logging_config.py (5: null read, group guard, write/read/clear roundtrip, expired-null, invalid-input 400s).

- Real-device observations are retained as supporting engineering evidence only. The submission does not claim a separate formal hardware-validation study; any future physical claim requires a controlled protocol, acceptance criteria, retained operator record, and evidence disposition.

TV-ALG-010 uses the thresholds stated in its verification row above. The controlled STR must retain the baseline identity, execution environment, executor, checksums, coverage reports, and an exception package for every shortfall. The exact-freeze results and controlled locators are summarized in Appendix A07, the Formal Evidence Index.

5.1 Deterministic Simulation Campaign

This campaign adds deterministic scenario replay (medium-fidelity mocks) as a required verification layer for runtime safety logic. It complements hardware-in-the-loop testing and does not replace real-device validation.

Test ID	Level	Purpose	SRS Link
TV-SIM-001	Integration (deterministic sim)	Reproduce anchored cadence across reconnect/relaunch windows and assert step index continuity (expected, executed, skipReason)	SRS-RUN-001, SRS-RUN-002, SRS-STATE-001
TV-SIM-002	Integration (deterministic sim)	Validate step-0 hard gate and step>0 degraded CGM execution (-1) across stale/out-of-range/noisy sensor sequences	SRS-CGM-001, SRS-CGM-002, SRS-CGM-003
TV-SIM-003	Integration (deterministic sim)	Validate pump-unknown/unavailable execution with command-block and no	SRS-PUMP-001, SRS-PUMP-005

Test ID	Level	Purpose	SRS Link
		false delivery application	
TV-SIM-004	Integration (deterministic sim)	Validate meal and BG trigger interplay under missed-step, reconnect, and degraded-input conditions	SRS-MEAL-001, SRS-MEAL-002, SRS-BG-002, SRS-BG-003
TV-SIM-005	Integration (deterministic sim)	Validate alert lifecycle, countdown refresh progression, dedupe, and clear behavior during state churn	SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-010
TV-SIM-POD-001	Integration (deterministic pod simulation)	Validate stateful pod ledger math for bolus progress/cancel, fallback mask expiry/renewal, pod expiry, and pod replacement identity continuity without live pod burn	SRS-PUMP-006, SRS-PUMP-008, SRS-RUN-006
TV-SIM-POD-002	Integration (deterministic pod simulation)	Validate pending issued-dose restoration and replay through simulated pump status using production runtime coordinator logic	SRS-STATE-005, SRS-MEAL-005, SRS-LOG-009
TV-SIM-POD-003	Integration (deterministic pod simulation)	Validate that meal/correction delivery completed before the fallback baseline is not subtracted from	SRS-PUMP-008, SRS-PUMP-009, SRS-LOG-009

Test ID	Level	Purpose	SRS Link
		fallback pump-total delta	
TV-SIM-POD-004	Integration (deterministic pod simulation)	Validate high-risk pod scenarios using shared invariants: - canceled and consecutive-canceled meal evidence feed - subsequent meal announcements without replay or stale evidence reuse, force-quit/relaunch restores in-flight meal delivery knowledge before the next CGM, relaunch with pending fallback replay after step 0 can replay with unavailable CGM inputs without dropping the plan, lost final meal response before status refresh is preserved until same-pod reconnect, same-pod completed meal and reservoir-capped partial meal evidence feed - the live attribution step without replay, different/new-pod, user-escaped unavailable-pod, and unresolved correction-	SRS-PUMP-008, SRS-PUMP-009, SRS-PUMP-010, SRS-MEAL-005, SRS-LOG-009

Test ID	Level	Purpose	SRS Link
		only/basal-only issued doses are assumed delivered without blocking replacement-pod dosing when policy allows, stale cached idle is not trusted when fresh refresh fails, fallback maintenance block prevents the live bolus command, meal/fallback delta partitioning separates issued dose from fallback residual, fallback replay allocation follows the programmed schedule weights only after mask expiry, reservoir- capped fallback replay uses observed pump delta instead of modeled exposure, ambiguous fallback restore does not replay or inject modeled fallback insulin into the live step, assumed old-pod fallback replay emits no pump commands and allows replacement-pod live dosing, missing- status/nonreplaya	

Test ID	Level	Purpose	SRS Link
		ble fallback plans are cleared without row emission, matching pending meal-progress state clears when issued-dose evidence is consumed, and app recent-dose rows, local step CSV export, and runtime cloud step-event emission preserve/display merged replay evidence source, disposition, request step, requested units, and delivered units	

Controlled evidence:

- STR-SIM-* scenario reports with script file, expected output snapshot, actual output snapshot, and pass/fail deltas.
- Controlled simulation harness: emits run-context, results, trace-map, and suite logs.
- Risk-based merge gate: runs TV-SIM-* / pod-sim rows when high-risk runtime, pump, fallback, or reconciliation paths are touched.
- Exact-freeze result: Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/.

Future extension (high-fidelity):

- After medium-fidelity stability, add BLE/session-level emulation cases for hardware-specific transport faults and timing jitter that mock services cannot represent.

Current implemented deterministic simulation coverage:

- testTVSIM001_AnchoredCadenceAcrossReconnectAndRelaunch (TV-SIM-001)
- testTVSIM002_StepZeroGateAndStepGreaterThanZeroDegradedCGMExecution (TV-SIM-002)

- testTVSIM003_PumpUnavailableAndUnknownStatesBlockLiveExecution (TV-SIM-003)
- testTVSIM004_MealAndBGInterplayAcrossMissedStepsAndReconnectChurn (TV-SIM-004)
- testTVSIM005_AlertLifecycleChurnCountdownDedupeAndClearTransitions (TV-SIM-005)
- testSimulatedDashPodReportsBolusProgressAndCancelPartialDelivery (TV-SIM-POD-001)
- testSimulatedDashPodAccruesFallbackBasalOnlyAfterMaskExpires (TV-SIM-POD-001)
- testSimulatedDashPodDoesNotExposeFreshFallbackDeltaWhileDisconnected (TV-SIM-POD-001, TV-SIM-POD-003)
- testSimulatedDashPodMaskRenewalExtendsZeroBasalSuppression (TV-SIM-POD-001)
- testSimulatedDashPodExpiryStopsDeliveryAndReportsUnknownState (TV-SIM-POD-001)
- testSimulatedDashPodReplacementResetsCountersAndRejectsOldPodAttribution (TV-SIM-POD-001)
- testSimulatedDashPodReplacementClearsSuspendedState (TV-SIM-POD-001)
- testSimulatedDashPodCapsBolusDeliveryAtReservoirRemaining (TV-SIM-POD-001)
- testScriptedPodPumpServiceDoesNotReportStartForPreStartBolusFailure (TV-SIM-POD-001)
- testSimulatedDashPodRestoresPersistedPendingDoseFromPastRequestTime (TV-SIM-POD-002)
- testSimulatedDashPodDoesNotAdvanceClockForFuturePendingAttributionRestore (TV-SIM-POD-002)
- testScriptedPodPumpServiceFeedsCompletedIssuedDoseIntoReplayWithoutHardwareFixtures (TV-SIM-POD-002)
- testCompletedMealBeforeFallbackBaselineDoesNotPolluteFallbackDelta (TV-SIM-POD-003)
- testMealDoseOverlappingFallbackWindowPartitionsPumpDeltaBeforeFallbackReplay (TV-SIM-POD-003, TV-SIM-POD-004)
- testPodScenarioCanceledMealEvidenceFeedsNextMealAnnouncementWithoutReplay (TV-SIM-POD-004)
- testPodScenarioConsecutiveCanceledMealsConsumeLatestEvidenceWithoutReplay (TV-SIM-POD-004)
- testPodScenarioForceQuitRelaunchRestoresInFlightMealBeforeNextCGM (TV-SIM-POD-004)
- testPodScenarioLostFinalMealResponseDisconnectsBeforeRefreshThenReconcileOnSamePodReconnect (TV-SIM-POD-004)
- testPodScenarioSamePodCompletedMealFeedsLiveAttributionStepWithoutReplay (TV-SIM-POD-004)

- testPodScenarioReservoirCappedMealFeedsActualDeliveredUnitsWithoutReplay (TV-SIM-POD-004)
- testPodScenarioDifferentOrNewPodAssumesIssuedDoseAndAllowsLiveDose (TV-SIM-POD-004)
- testPodScenarioReplacementAfterUnreconciledOldPodDoseAssumesOldDoseAndAllowsNewPodLiveDose (TV-SIM-POD-004)
- testPodScenarioUserEscapedUnavailableOldPodConsumesAssumedEvidenceAndAllowsNewPodDose (TV-SIM-POD-004)
- testPodScenarioUnresolvedCorrectionWithoutPodIdentityAssumesDeliveredAndAllowsLiveDose (TV-SIM-POD-004)
- testPodScenarioUnresolvedBasalMicrodoseAssumesDeliveredAndAllowsLiveDose (TV-SIM-POD-004)
- testPodScenarioStaleCachedIdleDoesNotAuthorizeBolusWhenFreshRefreshFails (TV-SIM-POD-004)
- testPodScenarioFallbackMaintenanceBlockPreventsLiveBolus (TV-SIM-POD-004)
- testPodScenarioFallbackReplayUsesScheduleWeightsOnlyAfterMaskExpiry (TV-SIM-POD-004)
- testPodScenarioReservoirCappedFallbackReplayUsesObservedPumpDelta (TV-SIM-POD-004)
- testPodScenarioMealOverlapPartitionsPumpDeltaAndMergesIntoFallbackReplay (TV-SIM-POD-004)
- testPodScenarioPendingFallbackReplayRunsOnRelaunchWithoutFreshCGMAfterStepZero (TV-SIM-POD-004)
- testPodScenarioAmbiguousFallbackRestoreDoesNotReplayOrInjectModeledDose (TV-SIM-POD-004)
- testPodScenarioAssumedFallbackReplayFromUnavailableOldPodAllowsReplacementDose (TV-SIM-POD-004)
- testPodScenarioNonReplayableFallbackPlanClearsWithoutReplayOrModeledDoseInjection (TV-SIM-POD-004)
- testRecentDoseTimelinePersistsMergedMealFallbackReplayEvidenceLabels (TV-SIM-POD-004)
- testCSVExportIncludesFallbackReplayRows (TV-SIM-POD-004, TV-LOG-002, TV-LOG-009)
- testEmitExecutionTelemetryPublishesReplayAndLiveAlgorithmStepSnapshots (TV-SIM-POD-004, TV-LOG-009)

Current implemented alert-test coverage:

- testTopAlertPrefersHigherSeverityThenMostRecent and testSortedAlertsOrdersBySeverityRecencyAndStableDedupeKey cover deterministic alert ordering precedence (TV-ALERT-002 subset).

- `testHomeSurfaceHidesSteppingInterruptedWhileASpecificPauseExplainsIt`, `testHomeSurfaceHidesCommandBlockedDuringPumpOutage`, `testHomeSurfaceNeverSuppressesSafetyCriticalAlerts`, `testHomeSurfaceKeepsUnrelatedAlertsUntouched`, and `testHomeHiddenAlertCountReportsSuppressedAlertsSoHomeCanPointAtAlertCenter` cover the Home alert-stack root-cause suppression matrix: generic stepping-interrupted/command-blocked alerts hidden only while a specific active alert explains them, safety-critical alerts never suppressed, unrelated alerts untouched, and the hidden-related-alert count that drives the Home "related alert in Alert Center" affordance so the bell badge and Home surface never silently disagree (TV-ALERT-010 subset).
- `testNoActivePodDebounceAddsAndClearsAlert`, `testNoActivePodConditionRepeatsBackgroundNotificationUntilRecovered`, and `testHomeAlertSyncEvaluatorReflectsCombinedPumpConditions` cover no-active-pod alert path, repeated safety-critical background notification attempts while the condition remains true, clear-on-recovery behavior, and suppression of competing signal-loss state when no pod is present (TV-ALERT-003, TV-ALERT-004, TV-ALERT-016 subset).
- `testSignalLossDebounceAddsAndClearsAlert` covers debounce + auto-clear behavior, actionable background notification cooldown/dedupe, and clear-on-retract notification cleanup (TV-ALERT-003, TV-ALERT-004, TV-ALERT-006 subset).
- `testSignalLossDebounceSuppressesTransientCondition` covers transient suppression and notification authorization priming dedupe (TV-ALERT-003, TV-ALERT-006 subset).
- `testShowPreviewAlertsSupportsMultipleTypesAndPrecedence` covers severity-filtered background notification routing (critical not informational), alert-category route mapping, and safety-critical acknowledge behavior (TV-ALERT-002, TV-ALERT-004, TV-ALERT-006 subset).
- `testCloudTelemetryReporterSurfacesSubjectIDConflictAndStopsRetryFor409Conflict`, `testHomeSettingsViewClearsResolvedSubjectIDConflictAlert`, `testSubjectIDConflictAutoResolutionPolicyRequiresActiveAlertNonEmptySubjectAndNoInFlightCheck`, and `testSubjectIDConflictAutoResolutionPolicyThrottlesSameSubjectAndAllowsChangedSubject` cover the app-policy subject-ID conflict alert lifecycle: issue on permanent cloud claim conflict, explicit retract after successful corrected Clinical Settings save, and throttled Home auto-revalidation of the currently persisted subject ID when a stale conflict alert remains active (TV-ALERT-005 subset).
- `testCGMAlertsNeverScheduleBackgroundNotifications`, `testCGMAlertMapperFailedFromSensorFailedState`, `testCGMAlertMapperUnavailableFromWarmupState`, and `testCGMFailedAlertRestoresFromLiveStateAcrossAlertCenterResetUntilRecovery` cover the CGM availability/failure policy: informational in-app status only, no required-ack path, and no background local notifications (TV-ALERT-014, TV-ALERT-006 subset).

- `testCGMUrgentLowAlertMapperIssuesForReliableReadingBelow55`, `testCGMUrgentLowAlertMapperClearsAt55OrAbove`, `testCGMUrgentLowAlertMapperSkipsUnreliableReading`, `testCGMUrgentLowAlertMapperSkipsStaleReading`, `testUrgentLowAcknowledgeMarksAlertReviewedWithoutClearingActiveState`, `testCGMUrgentLowAcknowledgePersistsAcrossAlertCenterResetUntilRecovery`, and `testCGMAlertsNeverScheduleBackgroundNotifications` cover the app-derived urgent-low review alert trigger, trustworthy-data gate, reviewed-state retention, reset/reattach persistence, recovery auto-clear, and no-OS-notification policy (TV-ALERT-015, TV-ALERT-006 subset).
- `testAlgorithmCheckBGRequestSurfacesDedupedHapticAlertAndClears`, `testAlgorithmCheckBGRequestDoesNotRestoreAfterRelaunch`, and `testLoopRuntimeEngineSurfacesAndClearsCheckBGRequestFromAlgorithmOutput` cover the Algo2015 checkBG nudge: prompt creation, haptic-on-new-request behavior, dedupe while active, clear on flag clear, runtime publication, and stale-relaunch suppression (TV-ALERT-017 subset).
- `testForcedOpenLoopOutputIsReachableAfterSustainedMissingCGM`, `testForcedOpenLoopOutputBlocksAutomatedPumpCommandAfterMaintenance`, and `testLoopRuntimeEngineSurfacesAndClearsForcedOpenLoopOutput` cover the Algo2015 forced-open-loop path: C++ reachability after sustained missing CGM, pre-command maintenance preservation, automated-command block, telemetry retention, safety-critical alert issue/retention across skipped publications, and explicit clear on a later non-forced-open algorithm output (TV-ALERT-018 subset).
- `testAlertCenterTracksRecentlyClearedAlerts` and `testAlertCenterRestoresPersistedActiveAndClearedAlerts` cover in-app Alert Center active/recent behavior and persistence restore path (TV-ALERT-007, TV-ALERT-008 subset).
- `testPumpAlertMapperExpiringIncludesCountdownDeadline`, `testPumpAlertMapperExpiredForPodExpiringAlert`, and `testTimeSensitivePumpExpiringAlertRefreshesMessageWithoutReschedulingNotification` cover pod-expiration countdown mapping (expiring and expired paths) plus minute-refresh text updates without extra background notification scheduling (TV-ALERT-009, TV-ALERT-006 subset).
- UI automation now covers Home-to-Alert-Center routing, acknowledge-to-recent flow, and relaunch persistence visibility (`testUI007_HomeAlertCenterButtonOpensAlertCenter`, `testUI008_AlertCenterAcknowledgeMovesAlertToRecentlyCleared`, `testUI009_AlertCenterPersistsAcrossRelaunch`) (TV-ALERT-007, TV-ALERT-008 subset).
- `testPumpPersistedAlertStoreReturnsIssuedAndRetractedAlerts` and `testCGMPersistedAlertStoreReturnsIssuedAndRetractedAlerts` cover delegate `PersistedAlertStore` issue/retract lookup behavior (TV-ALERT-008 subset).

- `testPumpExpirationAlertSyncPlannerReturnsRetractsWhenNoExpirationAlertsApply` covers no-active-pod retract-set safety boundary by excluding critical fault/incompatible alerts from auto-retract cleanup (TV-ALERT-011 subset).
- `testCGMAlertMapperPrioritizesUnavailableOverFailedKeywordCollision` and `testCGMAlertMapperDoesNotClassifyMessageOnlyFailedAsSensorFailure` verify CGM fallback keyword mapping cannot escalate transient/message-only text into ALERT-CGM-FAILED-OR-EXPIRED (TV-ALERT-001 subset).

Current implemented runtime-refactor regression coverage:

- `testMealPumpUnavailableReasonMapping` verifies meal-unavailable reason precedence (noPump over signalLoss when no active pod exists) (TV-MEAL-005 subset).
- `testMealAnnouncementSheetLifecycleRevalidatesOnlyOnForeground`, `testHomeRuntimeActionCoordinatorMealComposerContinuationDecision`, `testMealComposerRevalidationRearmsInPlaceWhenSlotConflictButFreshCheckIsAvailable`, `testMealComposerRevalidationShowsInlineUnavailableUsingFreshReasonAfterSlotConflict`, `testMealComposerRevalidationShowsInlineUnavailableForNonConflictReasons`, `testMealComposerRevalidationPrefersCancelDeliveryWhenPumpDeliveringWithContext`, and `testMealComposerRevalidationContinuesWhenSubmitAvailabilityIsAvailable` verify the foreground revalidation gate, the entry-time stale-composer remapping, and the open-composer revalidation policy: in-place re-arm on stale observed step with an available fresh check, inline blocked messaging using the fresh reason, cancel-delivery precedence, and continue-on-available (TV-MEAL-007 subset).
- `testMealComposerRevalidationShowsWaitingNoticeWhenBackupBasalWhileCarveOutBGConsumptionInFlight`, `testMealComposerRevalidationShowsGenericBackupBasalMessageWithoutCarveOutConsumption`, `testMealComposerRevalidationShowsWaitingNoticeForFreshBackupBasalAfterSlotConflict`, `testMealComposerRevalidationStillBlocksUnreconciledIssuedDoseWhenCarveOutBGConsumptionInFlight`, `testPumpDeliveringRevalidationReplacesTransientBGWaitingNotice`, `testReclosedLoopStepAfterCarveOutConsumptionRearmsComposerInPlace`, `testForcedOpenAvailabilityDuringInFlightCarveOutConsumptionRoutesToWaitingNotice`, `testForcedOpenAvailabilityAfterConsumingStepPublishedShowsGenericBackupBasalMessage`, `testMealComposerRevalidationRearmsFromConsumedCarveOutWhenFreshAvailabilityRecovers`,

- testMealComposerRevalidationContinuesFromConsumedCarveOutWhenSubmitAvailabilityIsAvailable,
 testMealComposerCarveOutBGTargetStepCapturedOnlyWithPendingValue,
 testMealComposerCarveOutBGConsumptionInFlightRequiresClearedValueAndUnpublishedQualifyingStep, and
 testMealComposerSubmitHoldsForCarveOutBGConsumptionOnlyWhenBackupBasalAndInFlight verify SRS-MEAL-006 v1.55: the transient waiting hold when a loop step has cleared the pending fingerstick but its qualifying-glucose marker has not published, including the field shape where a pre-command checkpoint already reports `lastExecutedStep == capturedTarget`; completion on the matching marker; immediate connected pump-busy presentation; generic backup-basal fallback after completed/nonqualifying passage; unreconciled-issued-dose isolation; recovery re-arm/continue; and the submit-time hold predicate. testSubmitAvailabilityPreservesHigherPriorityRuntimeBlocks additionally pins the backupBasalRunning passthrough (TV-MEAL-007 subset).
- testMealAnnouncementAvailabilityBlocksPersistedPendingMealRequestAcrossRelaunch,
 testMealAnnouncementAvailabilityReconcilesResolvedPendingMealRequestOnLaunch,
 testMealAnnouncementAvailabilityConsumesPersistedResolvedTelemetryReplayStateOnLaunch,
 testReconciledPendingMealAnnouncementStateClearsWhenTargetStepAlreadyExecuted,
 testMealAnnouncementResolutionEventUsesPersistedFlowIDForResolvedPendingState, and
 testMealAnnouncementResolvedEventUsesPersistedResolvedTelemetryReplayState verify persisted pending meal-request durability, relaunch duplicate blocking, replay-token consumption, target-step reconciliation, and correlated flow-ID closure for resolved lifecycle telemetry (TV-MEAL-009 subset, TV-LOG-007 subset).
 - LoopRuntimeCoordinatorMealAnnouncementDurabilityTests.testMealAnnouncePersistsPendingMealOnlyAfterExecutionStepAccepted and
 LoopRuntimeCoordinatorMealAnnouncementDurabilityTests.testMealAnnounceRejectedBeforeAcceptanceDoesNotPersistPendingMealState verify that pending meal state is written only after the coordinator has accepted a concrete execution step and is not left behind for rejected meal attempts (TV-MEAL-009 subset).
 - testAnnounceMealReturnsBlockedWhenLoopIsOff,
 testAnnounceMealReturnsBlockedWhenPersistedPendingMealExists,
 testReconciledUncertainPendingMealAnnouncementStateClearsWhenPumpDeliveryMatchesTargetStep,
 testMealAnnouncementResolutionEventUsesReconciledAfterUncertainForUncertainClear,
 testMealAnnouncePersistsPendingMealOnlyAfterExecutionStepAccepted,
 testMealAnnounceRejectedBeforeAcceptanceDoesNotPersistPendingMealState,

testMealAnnounceUncertainDeliveryRetainsPendingMealState, and
 testHomeMealAnnouncementSubmitPolicyBuildsSubmissionContext,
 testHomeMealAnnouncementSubmitPolicyEmitsLifecycleEvents,
 testHomeMealAnnouncementSubmitPolicyRequiresFlowIDForLifecycleEvents verify
 that meal submit no longer reports optimistic success, that blocked runtime outcomes map to
 explicit blocked results, and that Home/runtime expose deterministic
 submitted/accepted/success/uncertain/resolved telemetry closure with explicit
 uncertain reconciliation semantics (TV-MEAL-008, TV-MEAL-010, TV-LOG-007 subset).

- testHomeRuntimeActionCoordinatorRoutesActiveMealDeliveryToCancelDeliveryFlow,
 testMealAnnouncementCancelledDeliverySummaryUsesPartialDeliveryCopy,
 testMealAnnouncementCancelledDeliverySummaryHandlesNoDeliveredInsulin,
 testMealAnnouncementCancelledDeliverySummaryIncludesCancelDetails,
 testMealAnnouncementCancelledDeliveryPolicyRequiresFiveMinutesAndNextStep,
 testMealAnnouncementCancelledDeliveryPolicyUsesNextStepThreshold,
 testMealAnnouncementDisplaySupportMapsMealContext,
 testPumpServiceAdapterCancellationDeliveryStatusUsesRequestedAndDelivered
 Units,
 testPumpServiceAdapterCancellationDeliveryStatusInfersPartialWhenDelivered
 UnitsAreMissing,
 testPumpServiceAdapterResolvedBolusDeliveredUnitsPrefersPodCompletionWhen
 EventHistoryLags,
 testPumpServiceAdapterResolvedBolusDeliveredUnitsUsesBestAvailableProgress
 WhileBolusing,
 testPumpServiceAdapterAuthoritativeCompletedDeliveryPrefersCanceledUnitsW
 henIdle, testCanceledMealDeliveryRecordsEvidenceAndClearsPendingMealBlock,
 testMealAnnouncementAvailabilityAllowsPersistedMealAttributionWithMatchin
 gEvidence,
 testMealDeliveryProgressRestorePolicyDoesNotRestoreAfterMatchingEvidenceI
 sStored, testMealDeliveryProgressRestorePolicyRequiresMatchingPodIdentity,
 testRecordDoWorkResultMarksSuccessfulBolusAsDeliveringBeforePumpRefresh,
 testReconcilePumpStatusUpdatesInterruptedDeliveryToCompletedAfterLaterRef
 resh,
 testReconcileCanceledDeliveryUsesDeliveredUnitsForInterruptedMealBar,
 testPumpStatusObserverRefreshReconcilesSharedTelemetryStoreUntilDeliveryC
 ompletes,
 testPumpStatusObserverApplyCanceledBolusDeliveryReconcilesSharedTelemetry
 ,
 testInsulinChartPointFlagsInterruptedDeliveryWhenDeliveredLessThanRequest
 ed, testInsulinChartPointDoesNotFlagActiveDeliveryAsInterrupted,
 testInlineInsulinChartStylingUsesCautionColorForInterruptedDelivery,
 testInlineInsulinPointCompactorPreservesDeliveringStateWhenCollapsingPoin
 ts,

`testHomeViewStateBuilderActiveMealDeliveryCancellationContextUsesOnlyDeliveringMealStep`, and
`testUI002b_MealCancelDeliveryFlowShowsPartialDeliverySummaryAndComposer`
 verify the meal cancel-delivery path: active-delivery routing into a destructive Home inline cancel flow, automatic visibility of the cancel control while a meal bolus is still actively delivering, requested/delivered-unit reporting after cancellation, orange partial-delivery context in Home's alert-summary region above the chart, cancel-time plus meal-context summary detail, optimistic active-delivery chart state immediately after a successful bolus command, explicit canceled-delivery reconciliation into shared step telemetry so interrupted bar height matches actual delivered insulin, normal meal-color chart rendering while delivery is still active, compactor preservation of delivering state when bars visually collapse, caution-color rendering only for actual interrupted delivery derived from requested-vs-delivered telemetry, later pump-refresh reconciliation back to completed delivery when the bolus finishes normally, pod-status flooring when event-history delivery lags, immediate persisted cancellation evidence for runtime meal-unblock/replay accounting, restore suppression only for attribution/evidence with matching pod identity, and preservation of delivered insulin accounting for the next algorithm step when the operator later reopens meal announce (TV-MEAL-012 subset, TV-PUMP-003 supporting coverage).

- `BionicLoopMealAnnouncementRuntimeTests` and `BionicLoopMealAnnouncementPodReplacementEscapeTests` cover `testMealDeliveryPodReplacementEscapePolicyRequiresMealAndElapsedCompletionWindow`, `testMealDeliveryPodReplacementEscapePolicyAllowsKnownUnavailablePodWithoutWaitingForGrace`, `testMealDeliveryPodReplacementEscapePolicySuppressesEscapeAfterMatchingEvidence`, `testUserAbandonedMealDeliveryRecordsAssumedDeliveredEvidenceForPodReplacement`, and `testUserAbandonedMealDeliveryDoesNotResolveNonMealIssuedDose`, verifying the explicit user-confirmed pod-replacement escape: availability only for matching meal-linked attribution, timeout/known-unavailable gating, suppression after matching evidence, assumed-delivered evidence persistence with `user_abandoned_unavailable_pod`, meal-progress field clearing, issued-dose attribution preservation for replay/live accounting, and correction-only suppression (TV-MEAL-013, TV-PUMP-009 supporting coverage).
- `testCorrectionDeliveryCreatesPendingIssuedDoseAttribution`, `testPendingIssuedCorrectionDoesNotAdvanceWhilePumpStillDelivering`, `testFreshIdleStatusAfterPhysicalCompletionConsumesPendingIssuedDose`, `testThreeUnitBolusLeavesDeliveringBeforeNextFiveMinuteStep`, `testDisconnectedPodDoesNotReportDeliveringDuringBolusOutage`, `testReconciledCorrectionDeliveryReplaysFirstMissedPostDoseStepBeforeLiveResume`, `testReconciledMealDeliveryReplaysFirstMissedPostMealStepBeforeLiveResume`,

```

testIssuedDoseAttributionWithNonCredibleDeliveredUnitsAssumesRequestedDoseAndReplays,
testFallbackBasalExposureReconcilerPartitionsIssuedDoseFromPumpDeltaBeforeCredibilityCheck,
testFallbackBasalExposureReconcilerRejectsPendingIssuedDoseWhenPumpDeltaCannotCoverRequest,
testFallbackBasalExposureReconcilerRejectsPartitionWhenResidualDoesNotMatchFallbackExposure,
testFallbackBasalExposureReconcilerAcceptsLowerResidualAfterIssuedDosePartition,
testFallbackBasalExposureReconcilerPartitionsPersistedIssuedDoseEvidence,
testPartitionedIssuedDoseEvidenceReplaysWhenPumpLastDeliveryIsUnavailable,
,
testPartitionedIssuedDoseEvidenceMergesWithFallbackReplayWhenPumpLastDeliveryIsUnavailable,
testPartitionedIssuedDoseEvidenceFeedsLiveFirstAttributionStepWhenReplayIsNotRequired,
testReconnectRecoveryPartitionsPendingIssuedDoseBeforeFallbackReplayPlanning,
testReconnectRecoveryUsesLowerFallbackResidualAfterIssuedDosePartition,
testAmbiguousPartitionCarriesPumpTotalWithoutClaimingFallbackActualDelivery,
testRecentDoseTimelineFallbackReconnectRecoveryTextDoesNotClaimResidualForPumpTotalOnly,
testMealDeliveryAttributionMergesWithEchoedAskWhenNoPreFallbackRoomExists,
,
testMealDeliveryAttributionPrecedingFallbackReplaysPureConfirmRowBeforeFallbackRows,
testMealDeliveryAttributionPrecedingFallbackRetractsSentinelAsksAcrossWiderDeadGap,
testPreActivationIssuedDoseReplaysConfirmRowAndSentinelsBeforeFallbackRows, testFallbackReplayPumpStatusDoesNotMergeNonCredibleIssuedDoseDelivery,
testMealDeliveryAttributionAssumesDeliveredWithoutMatchingPumpEvidence,
testMealDeliveryAttributionAssumesDeliveredWhenPumpEvidenceMismatchesRequest, testUnresolvedIssuedDoseAssumesDeliveredAndReplaysBeforeLiveStep,
testIssuedDoseAttributionWithoutPodIdentityAssumesDeliveredAndReplays,
testIssuedDoseAttributionAssumesDeliveredForDifferentOrNewPodAndReplaysBeforeLive,
testIssuedDoseAttributionAssumesDeliveredForDifferentOrNewPodEvenWhenObservedPodIsDelivering,
testPersistedAssumedDeliveredEvidenceKeepsClinicalPolicyReplayDisposition,
,

```

`testAssumedDeliveredDifferentOrNewPodFallbackMergeScrubsReplacementPodDeliveryFromLiveStep,`
`testAssumedDeliveredDifferentOrNewPodAllowsFutureAutomaticInsulinCommand,`
`testRuntimeResolutionDismissesWhenMatchingPendingMealAttributionClears,`
`testRuntimeResolutionNeverDismissesSimulatedPreviewProgress,`
`testMealAnnouncementAvailabilityIgnoresLegacyDifferentNewPodHoldAcrossRelaunch,`
`testMealAnnouncementAvailabilityBlocksBlockingIssuedDoseAttribution,` and
`testBlockedMealAnnouncementAvailabilityMapsIssuedDoseReconciliationPending` verify generalized issued-dose attribution, active-delivery step blocking, the accepted DASH no-boundary-cancel liveness bound and fresh idle unblock path, same-request/same-pod missed-step replay, pump-total partition of known in-flight issued dose from fallback basal residual, acceptance of lower pump-reported fallback residuals after issued-dose subtraction, persisted partition evidence consumption on the live first attribution step when replay is not required, raw pump-total telemetry without claiming fallback residual in ambiguous/unresolved recovery, delivered-unit credibility rejection for over-modeled residuals, fallback replay overlap handling after mask expiry (pre-fallback confirm row + zero-delivery echo sentinels + seam ask echo when the attribution step precedes the fallback window; echoed-ask no-room merge otherwise), assumed-delivered-per-clinical-policy resolution of unresolved, mismatched, non-credible, and missing-identity evidence once a fresh settled post-request pump status exists (assumed-requested amount consumed by replay or live-step input), different/new-pod assumed-delivered attribution with replacement-pod live-step insulin-input scrubbing including fallback-replay merge overlap, retained clinical-policy replay disposition for persisted assumed-delivered evidence, future command allowance including automatic resume, meal progress modal resolution when runtime consumes a pending meal attribution without original-pod pump confirmation (with simulated preview progress - reachable only via UI-test launch arguments - exempt from runtime auto-resolution because it has no real delivery to observe), fallback-merge cleanup of matching pending meal-progress state, legacy hold nonblocking meal availability, and meal availability blocking/mapping (TV-PUMP-009, TV-STATE-004, TV-STATE-005).

- `testLoopRuntimeEngineResetAlgorithmSessionKeepsClinicalSettings` also confirms session reset clears runtime carry-over while preserving unrelated clinical settings; pending meal-request fields are included in that cleared runtime state (TV-MEAL-009 supporting coverage).
- `testDoWorkFeedsBackRequestedAndDeliveredWhenBelowDashMinimumQuantum` verifies delivery reconciliation preserves requested-vs-delivered values across steps when request is below DASH minimum deliverable quantum (TV-PUMP-003).
- `testLoopRuntimeWorkExecutorRecordsLatestReadingBeforeOperation,`
`testLoopRuntimeWorkExecutorSkipsRecordReadingWhenNoLatestReading,` and
`testLoopRuntimeWorkExecutorReturnsOperationResultWithoutMutation` verify behavior-preserving extraction for `doWork` execution snapshot sequencing.

- `testPumpBasalScheduleRejectsEntriesMissingMidnightAnchor`, `testPumpBasalScheduleRejectsNonIncreasingEntries`, `testPumpBasalScheduleReturnsCurrentRateForOffset`, `testPumpServiceAdapterMapsOmniBasalScheduleToCoreSchedule`, and `testPumpServiceAdapterMapsCoreBasalScheduleToOmniSchedule` verify the new domain-level programmed basal-schedule seam used by the masked offline-fallback subsystem: schedule validation at the core boundary, rate lookup behavior for a 24-hour repeating schedule, and adapter conversion between core `PumpBasalSchedule` and `OmniBLE` basal schedule persistence formats (TV-PUMP-004 supporting fallback coverage).
- `testPumpServiceAdapterBlocksProgrammedBasalScheduleReplacementDuringActiveTempBasal`, `testPumpServiceAdapterBlocksProgrammedBasalScheduleReplacementDuringActiveBolus`, and `testPumpServiceAdapterBlocksProgrammedBasalScheduleReplacementWhenBasalStateIsUnknown` verify that disruptive programmed-schedule replacement is rejected while a temp basal mask or bolus is active and also when basal-delivery state is unknown, so masked fallback arming/disarming cannot silently cancel active delivery or proceed without a confirmed steady-basal boundary (TV-PUMP-004 supporting fallback coverage).
- `testPumpServiceAdapterAllowsMaskedFallbackMaintenanceDuringZeroTempBasalMask`, `testPumpServiceAdapterBlocksMaskedFallbackMaintenanceDuringNonzeroTempBasal`, `testPumpServiceAdapterBlocksMaskedFallbackMaintenanceWhenBasalStateIsUnknown`, `testPumpServiceAdapterMapsLateMaskedFallbackBlockedReasonFromPumpManagerCommunicationError`, `testPumpServiceAdapterMapsPostScheduleMaskBlockedErrorToRemaskFailure`, and `testPumpServiceAdapterMapsPostSchedulePumpBlockToRemaskFailure` verify that the dedicated masked-fallback maintenance/disarm seam allows the expected connected zero-mask steady state while still blocking disruptive schedule replacement during unsafe or unknown pump-delivery conditions, normalizing late bolus-blocked renewal errors into deferred maintenance, and classifying post-schedule mask blocking/failure as recovery-required remask failure rather than a clean first-arm block (TV-PUMP-007, TV-PUMP-008).
- `testMaskedFallbackMaintenanceSkipsWithoutActiveAlgorithmSession`, `testMaskedFallbackMaintenanceArmsFallbackImmediatelyForNewSession`, `testMaskedFallbackMaintenanceProgramsFourBucketSafetyNominalScheduleWhenAvailable`, `testPrePumpCommandMaintenanceReceivesStepZeroFallbackCandidateBeforeBolus`, `testPrePumpCommandMaintenanceBlockReasonSuppressesPumpCommand`, `testPrePumpCommandFallbackArmCleanBlockAllowsPumpCommand`, `testPrePumpCommandFallbackArmRemaskFailureBlocksPumpCommandForRecovery`, and `testPrePumpCommandFallbackArmSkipsWhenPumpStatusUnavailable`,

- testPostExecuteFallbackMaintenanceSkipsSameCycleAfterPreCommandAttempt,
 - testPostExecuteFallbackMaintenanceSkipsUnavailablePumpStatus,
 - testPostExecuteFallbackMaintenanceRunsForAvailableStepWithoutPreCommandAttempt,
 - testPreExecutionMaskedFallbackMaintenanceArmsMissingFallbackBeforeNextStep,
 - testPreExecutionMaskedFallbackMaintenanceDefersArmWhenFreshPumpStatusUnavailable,
 - testMaskedFallbackArmBlockedLeavesFallbackUnarmedAndEmitsArmFailure,
 - testMaskedFallbackMaintenanceDoesNotRenewMaskBeforeEnteringTwentyMinuteWindow,
 - testPreExecutionMaskedFallbackMaintenanceRenewsMaskWithinTwentyMinuteWindow,
 - testPreExecutionMaskedFallbackMaintenanceDefersExistingRenewalWhenFreshPumpStatusUnavailable,
 - testPostExecuteMaskedFallbackMaintenanceDoesNotRenewExistingMaskedFallbackWithinWindow,
 - testPreExecutionMaskedFallbackMaintenanceDefersRefreshWhenScheduleWriteIsBlockedAndMaskStillActive,
 - testPreExecutionMaskedFallbackScheduleRefreshRemaskFailureRequiresImmediateReconciliation, and
 - testPreExecutionMaskedFallbackMaintenanceRecordsFailureWhenMaskRenewalIsUnacknowledged
- verify active-session gating, immediate programming when no fallback is currently armed, four six-hour fallback schedule programming from the secondary/safety q5 nominal profile, same-step arm-before-bolus behavior when step 0 or another current step first produces the fallback candidate, skipped pre-command maintenance when pump status is unavailable/unknown, fresh idle pump-status gating before pre-execution arm/renew/refresh maintenance, clean blocked first-arm behavior that leaves fallback unarmed while allowing the normal step command, same-cycle post-execution retry suppression after a pre-command fallback event, remask-failure behavior that enters reconciliation-required recovery and blocks the normal step command, pre-step arm-before-bolus behavior when a valid candidate already exists, explicit unarmed failure when the first arm is blocked, the 20-minute renewal window, pre-step renewal before the next loop execution, suppression of redundant post-step renewal for an already-masked fallback, deferred schedule refresh when a late blocked maintenance race occurs while the current mask is still active, and recovery-required handling for existing masked-fallback renewal or schedule-refresh remask failures (TV-RUN-008, TV-PUMP-007, TV-STATE-004, TV-LOG-009).
- testLoopRuntimeEngineResetAlgorithmSessionPreservesMaskedFallbackRecoveryContextWhenRestoreFails,
 - testLoopRuntimeEngineArmAlgorithmBlockedWhenMaskedFallbackRestoreIsPending,
 - testLoopRuntimeEnginePumpReconnectRecoveryAfterResetRestoresFallbackButKe

- epsLoopOff,
 testLoopRuntimeEngineStartResolvesPendingMaskedFallbackRecoveryByPreservingExistingSession,
 testLoopRuntimeEnginePumpReconnectRecoveryResumesExistingSessionEvenWhenLatestCGMReceiptIsFresh, and
 testLoopRuntimeEnginePumpReconnectRecoveryMarksConfirmedFallbackExposureWhenSnapshotMatchesFallbackSchedule verify failed restore recovery is preserved across reset, reconnect recovery honors explicit reset/loop-off intent instead of auto-rearming, delayed restore completion drives the persisted recovery timestamp/exposure window, successful reconnect preserves the existing session/cadence anchor, and confirmed/corrected basal-only recovery creates pending pump-delta reconciliation state plus modeled-vs-pump-reported exposure detail needed for missed-step algorithm replay (TV-PUMP-008, TV-STATE-004, TV-LOG-009).
- testPreExecutionMaskedFallbackMaintenanceDefersRenewalWhenBolusBlocksMaskAndMaskStillActive,
 testMaskedFallbackMaintenanceEmitsMaskExpiredWithoutRemaskingOnNoStepWake ,
 testLoopRuntimeEngineArmAlgorithmBlockedWhenMaskedFallbackReconciliationIsPending, and
 testLoopRuntimeEngineTriggerDoWorkPublishesBlockedResultWhenMaskedFallbackReconciliationIsPending verify that blocked renewal is treated as deferred while the current mask is still active, true mask expiry records reconciliation-required recovery state without immediate remasking, a new session cannot arm while that recovery is pending, and ordinary loop doWork is suppressed until recovery resolves (TV-RUN-008, TV-PUMP-007, TV-STATE-004, TV-LOG-009).
 - testLoopRuntimeEngineManualBGRunBlockedWhenMaskedFallbackReconciliationIsPending,
 testMealAnnouncementAvailabilityBlocksMaskedFallbackReconciliationAcrossRelaunch, and
 testMealAnnouncementPresentationFallbackReconciliationRequired verify that the same offline-expiry recovery state blocks manual BG execution and meal announce while surfacing reconnect/recovery-required guidance through user-facing availability messaging (TV-BG-011, TV-MEAL-006, TV-STATE-004, TV-UI-002).
 - testMaskedFallbackDisarmRestoresOriginalScheduleAndEmitsDisarmedEvent and testMaskedFallbackDisarmFailsWithoutActivePumpService verify reset/disarm restore behavior produces explicit success/failure review events instead of silently dropping the underlying programmed basal schedule state (TV-PUMP-008, TV-LOG-009).
 - testReconnectRecoveryAssumesModeledFallbackDeliveryForRetiredPodWithoutPumpEvidence,
 testFallbackReplayPlanDecisionAllowsAssumedDeliveredRetiredPodRecovery, and
 testLoopRuntimeEnginePumpReconnectRecoveryAssumesFallbackDeliveredWhenRet

`iredPodCannotRestore` verify that an inactive/retired or service-stopped old pod can resolve masked-fallback recovery by recording modeled fallback exposure as `assumed_delivered_per_clinical_policy`, queueing bounded no-command replay, clearing the recovery state, and preserving the existing algorithm session (TV-PUMP-008, TV-STATE-004, TV-LOG-009).

- `Q5NominalBasalProfileTests` and `LoopRuntimeCoordinatorNominalBasalProfileTests` verify q5 nominal-basal profile imputation after the first observation, local-time slot indexing, seven-observation slot history, four six-hour bucket generation, per-bucket observed-slot counts (`testObservedSlotCountsByBucketAreAllZeroForEmptyProfile`, `testFourBucketScheduleCarriesPerBucketObservedSlotCounts`, `testFullyObservedDayReportsFullPerBucketObservedSlotCounts`: empty profile reports all-zero counts, partial observation reports exact per-bucket counts on the generated schedule snapshot, full day reports 72 per bucket), and separate primary versus secondary/safety profile persistence from algorithm telemetry (TV-STATE-004, TV-PUMP-007, TV-LOG-009).
- `FallbackBasalExposureReconcilerTests`, `FallbackBasalReplayPlannerTests`, and `LoopRuntimeCoordinatorFallbackReplayTests` verify the core-owned basal-only reconnect evidence classification, replay-plan/pump-delta math (including scaling to pump-reported delivered insulin), coordinator replay of fallback-active missed primary/secondary algorithm steps in the bounded replay range with `CGM=-1`, per-step delivered insulin, explicit 0 U inputs for fallback-active missed steps without pump allocation, exclusion of pre-activation disconnected missed slots from replay, single-use cleanup of invalid/not-required/stale pending replay plans, and suppression of catch-up pump commands or duplicate aggregate live-step delivery (TV-PUMP-008, TV-STATE-004). The reconnect recovery suite now additionally verifies that reconnect schedule mismatch upgrades to a corrected reconciled result only when reconnect evidence is credible (fresh connected delivery baseline, delivery measurement that spans the outage, monotonic pod total, and close agreement between modeled and pump-reported delivered insulin), that stale or weak delivery evidence remains unreconciled, that missing pump-reported delivered-insulin delta prevents reconciliation except for the explicit retired/no-active-pod assumed-delivered policy path, that same-pod issued-dose evidence is subtracted from raw pod-total delta before fallback residual replay only when the pending dose is fully covered by the fallback baseline-to-recovery pump-total measurement window, that pre-baseline issued-dose timing remains unreconciled/no-replay, that lower pump-reported residuals after issued-dose subtraction are replayed as corrected fallback evidence while over-modeled residuals remain ambiguous, that historical CGM samples are not required for pump-delta reconciliation, that modeled fallback exposure integrates persisted multi-entry fallback schedules across bucket boundaries, and that reconnect diagnostics always log the persisted fallback schedule, the cached programmed schedule from manager state, the original programmed schedule, and the cached schedule source for later CloudWatch review.

- `testDoWorkDoesNotTrustCachedIdlePumpStatusWhenRefreshFails` verifies that a stale cached idle pump status cannot allow recommendation or command application when a fresh status refresh fails, reinforcing that runtime pump availability is determined by the current refresh result rather than UI/cache state (TV-PUMP-001, TV-RUN-008 supporting coverage).
- `testLoopSessionStorePersistsAlgorithmArmedAndRuntimeState` and `testLoopSessionStoreClearRuntimeStateReturnsEmptyState` verify session persistence boundaries.
- `testLoopWorkSchedulerOnlyTriggersForNewTimestampWhileArmed` verifies CGM timestamp dedupe/arm/reset behavior.
- `testLoopAlertMediatorReportsSignalLossUntilKnownRefresh` and `testLoopAlertMediatorKeepsSignalLossForUnknownRefresh` verify signal-loss policy mediation behavior.
- `testRecordDoWorkResultStoresSecondaryAlgorithmTelemetry`, `testUserDefaultsLoopRuntimeStateStorePersistsLatestFallbackBasalEvent`, `testResetSessionClearsTelemetry`, `testCloudTelemetryReporterSendsStructuredFallbackEventPayload`, `testLoopTelemetryStoreEmitsFallbackCloudTelemetryForNewAndChangedEventsOnly`, `testRecentDoseTimelineIncludesPersistedFallbackEventsAheadOfOlderSteps`, `testRecentDoseTimelineFallbackMetadataIncludesSourceTargetDurationAndStatus`, `testRecentDoseTimelineFallbackScheduleUpdateTextIncludesRateMaskAndNextRefresh`, `testRecentDoseTimelineFallbackUnchangedScheduleTextIncludesRetainedRateMaskAndNextRefresh`, `testRecentDoseTimelineFallbackReconnectRecoveryTextIncludesExposureAndUnreconciledResume`, `testRecentDoseTimelineFallbackReconnectRecoveryTextKeepsLoopOffAfterExplicitReset`, `testRecentDoseTimelineFallbackReconnectRecoveryTextIncludesReplayDetails`, `testRecentDoseTimelineStepSummariesIncludePrimaryAndSecondaryAlgorithmDetails`, `testRecentDoseTimelineFallbackFailureTextIncludesKindAndReason`, `testRecentDoseTimelineFallbackRestoreTitlesCoverDisarmLifecycle`, `testUnchangedScheduleCheckEventCarriesScheduleAndProfileCoverageWithoutBulkRates`, `testArmedEventCarriesPerBucketObservedSlotCounts`, `testScheduleRefreshEventCarriesFullProfilePayloadWithPerBucketCounts`, `testRecentDoseTimelineFallbackScheduleRowsShowOutcomeLabels`, `testRecentDoseTimelineFallbackUnchangedRowShowsScheduleSegmentsCoverageAndProvenance`, and `testRecentDoseTimelineFallbackProvenanceMarksAllImputedAndSingleObservedBuckets` verify that fallback-basal review events can be persisted through runtime/user-default state, mirrored into the shared telemetry store/cloud emitter, cleared on session reset, and

rendered in Home Recent Dose Steps with source, target, duration, schedule-refresh update/unchanged detail, maintenance-deferred status, restore/remask failure detail, modeled-vs-pump-reported recovery detail, pump-delta reconciliation detail when confirmed/corrected recovery occurs, disarm lifecycle wording, per-step primary/secondary algorithm summaries, and fallback profile/schedule cloud payload fields for the masked offline-fallback subsystem, including the steady-cadence `schedule_checked_unchanged` payload (programmed schedule + coverage + per-bucket observed-slot counts, no bulk 288-rate array), per-bucket counts on arm/refresh events (`profile_observed_slot_counts_by_bucket`, pinned in `testCloudTelemetryReporterSendsStructuredFallbackEventPayload` and `testArmedEventCarriesPerBucketObservedSlotCounts`), and schedule-row outcome labels plus per-segment observed/imputed provenance with the below-50%-observed imputed marking (TV-LOG-009).

- `testCSVExportIncludesAlgorithmInputOutputHeadersAndRowValues` verifies the local step CSV export keeps stable schema and persisted algorithm input/output snapshots; masked-fallback reconnect recovery is now covered by coordinator, CSV, and runtime cloud emission tests that assert fallback-active missed-step algorithm replay rows are emitted from credible pump delta with `CGM=-1`, per-step delivered-insulin input, replay evidence source/disposition/failure-reason columns, including explicit 0 U algorithm inputs when no pump allocation belongs to a replayed fallback-active step and no synthetic replay rows for pre-activation disconnected gaps (TV-LOG-002, TV-LOG-009 subsets).

Current implemented clock-sync telemetry safety coverage:

- `testDeviceClockSyncMonitorFlagsSkewAndPublishesWarningAtThresholdBreach` verifies midpoint skew calculation and warning emission when absolute skew exceeds 600 seconds (TV-UI-008, TV-LOG-006 subset).
- `testDeviceClockSyncMonitorWithinThresholdReportsOKWithoutWarning` verifies `<=600s` skew reports ok and does not emit warning alerts (TV-UI-008 subset).
- `testDeviceClockSyncMonitorForegroundCheckUses24HourSuccessfulCheckGate` verifies foreground checks are gated by 24-hour successful-check interval (TV-LOG-006 subset).
- `testDeviceClockSyncMonitorTimezoneChangeForcesFreshCheckInsideForegroundGate` verifies timezone/time-change trigger bypasses the foreground gate and performs a fresh UTC check (TV-LOG-006, TV-UI-008 subset).
- `testDeviceClockSyncMonitorRetriesAndReturnsUnavailableWithoutWarningOnNetworkFailures` and `testDeviceClockSyncMonitorLimitsSkewWarningsToOncePer24Hours` verify retry/unavailable behavior and warning cooldown control (TV-UI-008 subset).

Current implemented CGM UI stale-display safety coverage:

- `testG7ViewModelMasksStaleReadingAndHidesTrendWhenTimestampOlderThanElevenMinutes` verifies stale CGM masking to -- and hidden trend arrow when reading age exceeds 11 minutes (TV-UI-007).
- `testG7ViewModelMasksUnreliableCurrentReadingAndDoesNotFallbackToHistoryValue` verifies unreliable current CGM readings are masked to --, trend is hidden, and UI does not fallback-display historical value while current state is unreliable (TV-UI-007).
- `testG7ViewModelMasksUnreliableCurrentReadingWithoutTimestampAndDoesNotFallback` verifies unreliable current reading masking remains enforced when `latestReadingTimestamp` is missing (restore/partial-state edge), preventing fallback numeric display (TV-UI-007).
- `testG7ViewModelMasksStalePersistedHistoryWhenNoLiveReadingExists` verifies stale persisted-history fallback is also masked to -- (TV-UI-007).
- `testG7ViewModelUsesFreshPersistedHistoryWhenLatestReadingIsUnavailable` verifies non-stale persisted-history fallback still displays glucose value (control case for TV-UI-007 boundary behavior).
- `testG7ViewModelDisplayFormattingMapsExtremeValuesToHighLow` verifies boundary formatting ($\leq 39 \rightarrow \text{LOW}$, $\geq 401 \rightarrow \text{HIGH}$) and unit-label suppression semantics for boundary text (TV-UI-009).
- `testInlineCGMChartDerivationDynamicYAxisMaximumAndValues` verifies stepped CGM y-axis scaling behavior (300/350/400) and corresponding tick derivation (TV-UI-010).
- `testInlineCGMChartUsesPointOnlyPresentation` verifies the CGM chart's connector stroke and connected area fill remain disabled while the dot layer remains the trace presentation (TV-UI-010).

Current implemented Algo2015 verification coverage:

- `Algo2015BridgeContractTests` methods cover initial bridge contract behavior for null-guard paths, state-reset edge handling (`stateData == nil && timeStep > 0`), subject-id nil/long boundary handling, and state handoff continuity (TV-ALG-001, TV-ALG-002, TV-ALG-003 baseline subset).
- `Algo2015GoldenVectorTests.testNominalCGMSequenceMatchesGoldenOutputs` locks a deterministic nominal replay vector for drift detection (TV-ALG-004 baseline subset).
- `Algo2015GoldenVectorTests.testUnavailableCGMSequenceProducesFiniteDeterministicOutputs` adds degraded/unavailable-CGM replay coverage (TV-ALG-005 baseline subset).
- `Algo2015GoldenVectorTests.testMealAndManualBGInputsProduceDeterministicMealPathSignals` adds meal/manual-BG intervention replay coverage (TV-ALG-006 baseline subset).
- `Algo2015GoldenVectorTests.testPersistedStateReloadMatchesContinuousExecution` and `Algo2015GoldenVectorTests.testResetToFreshStateProducesDeterministicStepZeroOutput` add persistence/reload/reset continuity verification (TV-ALG-007).

- `Algo2015GoldenVectorTests.testCGMBoundaryValuesRemainFiniteAndBounded` adds CGM boundary/sentinel replay coverage (TV-ALG-008 baseline subset).
- `Algo2015GoldenVectorTests.testHigherTargetProducesLessInsulinForSameHyperglycemicSequence` adds differential target-behavior verification (TV-ALG-009 baseline subset).
- `Algo2015OracleSupport` now provides a reusable oracle framework for deterministic replay, snapshot assertions, and continuity checks across Algo2015 test suites (TV-ALG-004, TV-ALG-005, TV-ALG-006, TV-ALG-007, TV-ALG-008).
- `Algo2015MetamorphicTests` adds property/metamorphic checks for deterministic replay identity and monotonic sensitivity to target/CGM transforms (TV-ALG-004, TV-ALG-009 supporting evidence).
- `Algo2015DifferentialReplayTests` adds staged differential replay with JSON report output (`differential-report.json`) and now asserts all pregnancy parameters are consumed (`targetMgDL`, `mealUpfrontPercent`, `tmaxMinutes`) with deterministic checks for target monotonicity, applied meal-upfront profile, and TMAX-driven output variation (TV-ALG-009).
- `Algo2015DifferentialReplayTests.testPregnancyDifferentialReplayProducesDeterministicReport` now additionally asserts that 90% upfront meal profile front-loads more meal insulin than 75% at meal step and in immediate post-meal cumulative window (TV-ALG-009).
- The exact-freeze STR-ALG-001 bundle executed coverage, input-field, core-requirement, differential, boundary-transfer, and static-analysis lanes.
- Frozen structural results are Algo2015 96.12% line / 87.77% branch and bridge 90.27% line / 65.85% branch. The controlled exception package records the threshold shortfalls and their required dispositions.
- The controlled bundle also contains immutable run context, checksums, assertion-to-SRS/TV mapping, analyzer output, branch-gap mapping, and a reproducibility recipe. Appendix A07 is the authoritative evidence locator.

6. Evidence

Expected evidence package per change:

- test command output (`xcodebuild, swift test`)
- failing/passing test IDs
- device test logs where applicable
- screenshots for UI safety behavior
- link to changed requirement and risk IDs

7. Deferred or Unclaimed Validation

The current package does not retain separate formal claims for Scout round trips, physical ZIP inspection, hardware fault-injection, or summative human-factors validation. Supporting observations may corroborate automated evidence but do not become formal evidence unless promoted through the controlled evidence process. A later retained claim requires a controlled protocol, acceptance criteria, execution record, and disposition.

Current automated coverage for CGM interruption behavior:

- `BionicLoopStepInterruptionAlertCenterTests.testAlgorithmSteppingInterruptionMonitoringSchedulesFutureNotificationAndRaisesAlertAtDeadline`
- `BionicLoopStepInterruptionAlertCenterTests.testAlgorithmSteppingInterruptionMonitoringClearsActiveAlertWhenSteppingResumes`
- `BionicLoopRuntimeEngineSessionInfrastructureTests.testLoopRuntimeEngineArmedSessionSchedulesStepInterruptionMonitoringAndResetClearsIt`
- `BionicLoopRuntimeEngineStepInterruptionInfrastructureTests.testLoopRuntimeEngineForegroundRefreshShowsStepInterruptionWhenThresholdExceededBeforeFirstStep`
- `BionicLoopRuntimeEngineStepInterruptionInfrastructureTests.testLoopRuntimeEngineForegroundRefreshUsesLastSuccessfulRunDeadlineWhenAvailable`
- `BionicLoopRuntimeEngineStepInterruptionInfrastructureTests.testLoopRuntimeEngineForegroundRefreshDoesNotShowStepInterruptionWhenDisarmed`

8. Xcode Automated UI Testing Strategy

Purpose:

- Use `XCTest` UI automation as repeatable verification evidence for deterministic UI behavior and requirement conformance.

Highest-value coverage areas:

- Navigation and modal routing correctness.
- Presence/enabled-state of safety-critical controls.
- State-to-message rendering for known inputs.
- Regression checks for setup flows and dismiss paths.
- Non-hardware-dependent interaction logic (for example meal sheet presentation/cancel behavior).

Not a primary tool for:

- BLE transport reliability and reconnect behavior.
- Background wake cadence and overnight timing reliability.

- Real pump delivery confirmation and physical device alert timing.

Execution model:

- Run UI tests on Simulator with deterministic launch fixtures.
- Use app launch arguments/environment to force reproducible runtime states.
- Use stable accessibility identifiers for controls, labels, and state badges.
- Keep one fast smoke suite as release gate; keep extended suite for nightly runs.

9. UI Automation Verification Mapping

- Automated UI evidence is acceptable for SRS-UI-* and portions of SRS-MEAL-* and SRS-ALERT-* where behavior is deterministic and fixture-driven.
- Hardware-coupled behavior is represented by supporting real-device observations unless a separate formal hardware claim is explicitly retained and approved.
- Preferred command:
 - `xcodebuild -project BionicLoop.xcodeproj -scheme BionicLoop -destination 'platform=iOS Simulator,name=iPhone 17' -only-testing:<UI-test-target> -parallel-testing-enabled NO test`
- Evidence artifacts:
 - test logs, pass/fail results, captured screenshots/attachments, and linked TV-* IDs in RTM.

Current Automated UI Suite Mapping

XCTest Method	TV-ID Link	Requirement Link	Notes
testUI001_HomeShowsPrimaryControls	TV-UI-005	SRS-UI-002	Smoke check for Home control availability using deterministic fixtures.
testUI001b_HomePrimaryActionsRemainVisibleWhileContentScrolls	TV-UI-005	SRS-UI-002	Expands a multi-alert Home stack, scrolls the reserved content region, and verifies Manual BG and Meal Announcement remain hittable at fixed screen coordinates.
testUI002_MealUnavail	TV-UI-002	SRS-UI-002	Verifies unavailable-state messaging path and dismissal UX.

XCTest Method	TV-ID Link	Requirement Link	Notes
ableWhenLoopOff			
testUI03_SettingsSheetCanDismiss	TV-UI-006	SRS-UI-002	Guards against modal navigation traps in settings entry path.
testUI04_ManualBGSheetCanOpenAndCancel	TV-UI-006	SRS-BG-001	Verifies explicit cancel path for manual BG entry UX.
testUI04b_ManualBGReviewRequiresExplicitAlertConfirmation	TV-BG-008	SRS-BG-001	Verifies native exact-value review, value-preserving Change, and explicit confirmed submission.
testUI04c_ManualBGRapidDoubleTapCannotSubmitWithoutAlertConfirmation	TV-BG-008	SRS-BG-001	Verifies repeated taps at the entry action cannot bypass the spatially separate confirmation.
testUI05_HomeShowsAlertBannerPreview	TV-ALERT-002	SRS-ALERT-003	Verifies deterministic top-alert preview rendering on Home.
testUI06_HomeShowsCr	TV-ALERT-002	SRS-ALERT-003	Verifies critical alert preview path and title rendering.

XCTest Method	TV-ID Link	Requirement Link	Notes
ititcalAlertPreview			
testUI07_HomeAlertCenterButtonOpenAlertCenter	TV-ALERT-007	SRS-ALERT-008	Verifies Home alert-center bell entry and active-alert visibility in Alert Center.
testUI08_AlertCenterAcknowledgesAlertToRecentlyCleared	TV-ALERT-007	SRS-ALERT-005 , SRS-ALERT-008	Verifies acknowledge transition from active alert state to recently-cleared timeline.
testUI09_AlertCenterPersistsAcrossRelaunch	TV-ALERT-008	SRS-ALERT-009	Verifies persisted active alert visibility after relaunch (UI_TEST_PRESERVE_DEFAULTS).
testUI021_SignedOutActiveTherapyKeepsLoginProminentAcrossHomeAndAlertCenter	TV-SEC-008	SRS-SEC-009	Verifies persistent Home login access during signed-out active therapy and the actionable Alert Center route back to authentication.
testUI022_SignedOutAccountAndSession	TV-SEC-008	SRS-SEC-009	Verifies signed-out Account & Session presents Log In rather than a misleading Log Out action and returns to authentication.

XCTest Method	TV-ID Link	Requirement Link	Notes
nOffers Login			
testUI010_ClinicalSettingsNavigatesAndKeepsUnlockUntilManualLock	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies clinical unlock success, local unlock persistence across Clinical Settings navigation within the unlock duration, and manual lock.
testUI011_ClinicalSettingsSaveDismissesSettingsSheet	TV-CLIN-009	SRS-CLIN-007, SRS-CLIN-008	Verifies Save+OK closes settings flow after review-confirmation path.
testUI012_ClinicalSettingsInvalidCodeBlocksUnlock	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies invalid unlock-code path shows explicit error and keeps clinician controls hidden.
testUI013c_ClinicalUnlockAcceptsGroupedDigitsAndRejectsRepeatedCode	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies grouped/pasted-style numeric input is normalized and that a locally burned counter cannot be reused on the same device.
testUI013d_ClinicalUnlockUsesOffline	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies Clinical Settings unlock is satisfied by locally provisioned verifier material and does not depend on an online verification call.

XCTest Method	TV-ID Link	Requirement Link	Notes
eProvisionedVerifierMaterial			
testUI013_ClinicalControlsVisibleOnlyInsideUnlockedClinicalSettings	TV-CLIN-002	SRS-CLIN-003	Verifies relocated Start/Reset controls are absent in general settings and present only in unlocked Clinical Settings.
testUI014_RegularTargetChangeRequiresApprovalCaptureAndPersists	TV-CLIN-011, TV-CLIN-013	SRS-CLIN-011, SRS-CLIN-007	Verifies regular-settings target changes block until approval fields are completed, then persist into clinician-visible applied target state.
testUI015_ClinicalTargetPickerFollowsSelectedProfileRange	TV-CLIN-010	SRS-CLIN-009, SRS-CLIN-010	Verifies the clinician target picker only exposes the targets enabled by the selected Pregnancy/Standard profile.
testUI016_ClinicalProfileChangeNormalizesTargetAndPersists	TV-CLIN-012, TV-CLIN-013	SRS-CLIN-012, SRS-CLIN-009, SRS-CLIN-010	Verifies changing the clinician-selected profile snaps an inherited out-of-range draft target to the nearest allowed value and persists the normalized result.

Evidence reference:

- The controlled UI automation evidence is indexed in Appendix A07.

Current Clinical Unit Mapping

XCTest Method	TV-ID Link	Requirement Link	Notes
ClinicalUnlockVerifierTests	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies the backend contract HMAC vectors, ASCII/grouped-code normalization, non-ASCII digit rejection, wrong-subject rejection, same/lower counter rejection, lookahead rejection, and unsupported-version rejection.
BionicLoopClinicalUnlockRuntimeTests	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002	Verifies app-side secure-state semantics: verifier material installation, subject/material mismatch rejection, material refresh preserving accepted counter while clearing active unlock/lockout state, accepted counter persisted before success returns, repeated code rejected after burn, missing verifier material blocks unlock, failed attempts lock out temporarily, unlock expiration enforced locally, and manual-lock storage failure surfaced without falsely closing clinician controls.
testClinicalSettingsPolicyNormalizationAndDefaults	TV-CLIN-003, TV-CLIN-004, TV-CLIN-005	SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006	Verifies allowed-option enforcement and deterministic fallback defaults for target/upfront/TMAX selectors.
testClinicalSettingsSavePolicyAllowInitialUnsetConfigWithoutUnlock, testClinicalSettingsSavePolicyPrepareSaveReviewBlockedStates	TV-CLIN-001	SRS-CLIN-001, SRS-CLIN-002, SRS-CLIN-013	Verifies initial unset configuration can reach review without an unlock, while locked subsequent edits and invalid/no-change saves remain blocked with deterministic reasons/messages.

XCTest Method	TV-ID Link	Requirement Link	Notes
testClinicalSettingsSavePolicyPrepareSaveReviewBuildsChangedFieldList	TV-CLIN-009	SRS-CLIN-007	Verifies review model includes complete changed-field set for old/new clinical config diff.
testClinicalSettingsSavePolicySaveApplySemantics	TV-CLIN-009	SRS-CLIN-007, SRS-CLIN-008, SRS-LOG-001	Verifies no persisted change before save confirmation, cancel preserves applied config, and saved config appears in next-step telemetry snapshot fields.
testClinicalSettingsSavePolicyUICriticalEvents	TV-LOG-005	SRS-LOG-005	Verifies deterministic <code>ui.critical</code> event mapping and detail payload for <code>state_viewed/submit/cancel/blocked</code> paths.
testClinicalSettingsPolicyTargetRangeProfiles	TV-CLIN-010 , TV-CLIN-012	SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-012	Verifies Pregnancy/Standard profile subsets and nearest-allowed normalization behavior when the active profile changes.
testRegularTargetChangeApprovalPolicyPrepareAndValidate	TV-CLIN-011	SRS-CLIN-011	Verifies participant target changes require approver name and approval timestamp before apply.
testRegularTargetChange	TV-CLIN-010 , TV-CLIN-011	SRS-CLIN-010, SRS-CLIN-011	Verifies participant target-change flow rejects no-op requests and targets outside the clinician-selected profile.

XCTest Method	TV-ID Link	Requirement Link	Notes
geApprovalPolicyBlocksNoChangeAndOutOfProfileSelection			
testRegularTargetChangeApprovalTelemetryEvents	TV-LOG-008	SRS-LOG-008	Verifies participant approval-capture telemetry includes target profile, requested/current target, approver name, and approval timestamp.

Current regression command used in development for this slice:

- `xcodebuild -project BionicLoop.xcodeproj -scheme BionicLoop -destination 'platform=iOS Simulator,name=iPhone 17' -only-testing:BionicLoopTests -parallel-testing-enabled NO test`

UI execution note for this slice:

- The BionicLoop UI-test target is wired into the current scheme and targeted UI cases can be launched with `xcodebuild ... -only-testing:<UI-test-target>/<UI-test-case> test`.
- Focused UI verification for `testUI014_RegularTargetChangeRequiresApprovalCaptureAndPersists`, `testUI015_ClinicalTargetPickerFollowsSelectedProfileRange`, and `testUI016_ClinicalProfileChangeNormalizesTargetAndPersists` passed on 2026-03-25; the controlled result is indexed in Appendix A07.
- Local simulator/xctranner instability may still require rerunning the focused UI lane in future environments, but this slice now has a captured green UI pass.

10. Manual Screenshot UI Review Protocol

Scope:

- Required for all user-facing changes, especially safety-state messaging, alert presentation, and clinical controls.

Capture set:

- Light mode and dark mode screenshots.
- Changed screen in: baseline state, interactive state, blocked/error state, and post-action state.
- If applicable, include one large-text (Dynamic Type) capture for key screens.

Review rubric:

- Typography and text integrity:
 - no clipping, truncation, overlap, or ambiguous wording.
 - units/values formatting is consistent (mg/dL, U, %, min, timestamps).
- Spacing and alignment:
 - consistent spacing rhythm and card/control alignment.
 - safe-area compliance; no accidental edge clipping.
- Visual hierarchy:
 - critical safety states and primary actions are immediately distinguishable.
 - secondary text does not compete with critical signals.
- Accessibility and contrast:
 - sufficient contrast in both themes.
 - color is supplemented by text/icon/position cues.
 - tappable controls remain legible and touch-accessible.
- Motion and transitions:
 - state transitions are smooth and non-jarring.
 - no stale labels/icons during animated or async state changes.

Evidence and traceability:

- Save screenshots and review notes under the applicable STR-* evidence path.
- Link accepted evidence through the controlled RTM, formal evidence index, and applicable anomaly or deviation record.

Requirements Traceability Matrix (RTM)

Status: Build 843 verification traceability approved for IDE submission Version: 1.86 Owner: BionicLoop engineering Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, Build 843 traceability acceptance and final software-package approval, controlled email dated 2026-09-07 EDT Baseline freeze SHA: 91c0e98a9bc9429a0486bebdebf fc7d8dbbe300e Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-04-05	Engineering	Initial controlled RTM draft
0.9	2026-04-06	BionicLoop engineering	Added controlled-document metadata and refined RA-009 cybersecurity trace mapping for the software-only package.
0.91	2026-04-07	BionicLoop engineering	Narrowed RA-009 to the current local-security claim set, aligned RA-013 and RA-015 evidence notes with the implemented baseline, and changed high-risk freeze blockers to Rerun needed status
0.92	2026-04-08	BionicLoop engineering	Added masked offline-fallback maintenance/disarm trace links.
0.93	2026-04-08	BionicLoop engineering	Clarified masked-fallback renewal-window, deferred-maintenance, and restore-context trace links.
0.94	2026-04-14	BionicLoop engineering	Added trace notes for reconciliation-required blocked state after offline mask expiry
0.95	2026-04-14	BionicLoop engineering	Added trace notes for reconnect restore/disarm retry and explicit fresh re-arm after successful masked-fallback recovery
0.96	2026-04-14	BionicLoop engineering	Updated reconnect recovery trace notes for fresh-session auto-arm and preserved fallback review history after successful masked-fallback recovery
0.97	2026-04-14	BionicLoop engineering	Tightened reconnect recovery trace notes so explicit reset/loop-off keeps the loop off after restore and recovery timing reflects restore completion
0.98	2026-04-14	BionicLoop engineering	Updated reconnect recovery trace notes to show same-session unreconciled resume on the current due slot after successful masked-fallback restore
0.99	2026-04-15	BionicLoop engineering	Added trace notes for modeled-vs-pump-reported fallback review, pod total-delivery baseline persistence, and pre-step missing-fallback arm behavior

Version	Date	Author	Summary of Changes
1.00	2026-04-15	BionicLoop engineering	Added trace notes for confirmed/corrected same-session basal-only replay, persisted replay-plan state, and replay-detail Recent Dose Steps coverage
1.01	2026-04-15	BionicLoop engineering	Added trace notes for dedicated <code>loop.fallback.event</code> cloud telemetry emission and fallback-event payload verification for BionicScout contract support
1.02	2026-04-15	BionicLoop engineering	Added trace notes for replayed-step propagation into local per-step telemetry / CSV export with explicit replay markers
1.03	2026-04-27	BionicLoop engineering	Added trace notes that fallback replay requires pump-reported delivered-insulin delta and does not substitute modeled expected delivery when pump reconciliation is unavailable
1.04	2026-05-06	BionicLoop engineering	Added trace notes for safety q5 nominal-basal profile persistence, four-bucket fallback schedule programming, schedule-aware exposure modeling, and fallback profile/schedule cloud telemetry
1.05	2026-06-02	BionicLoop engineering	Updated trace notes for pump-delta fallback missed-step algorithm replay and repeated no-active-pod critical alert verification
1.06	2026-06-06	BionicLoop engineering	Added trace notes for same-pod continuity replay gating, different/new pod no-replay continuation, pod-continuity telemetry, and explicit-operator-only session lifecycle
1.07	2026-06-08	BionicLoop engineering	Added trace notes for persisted-schedule-weighted pump-delta replay allocation and zero-weight replay suppression
1.08	2026-06-10	BionicLoop engineering	Added trace notes for current-step masked-fallback first-arm ordering before same-step pump command application
1.09	2026-06-11	BionicLoop engineering	Clarified that confirmed/corrected fallback replay is bounded to fallback-active missed slots and excludes pre-activation disconnected gaps
1.10	2026-06-11	BionicLoop engineering	Added trace notes for fresh-status pre-execution fallback maintenance gating, existing-mask remask-failure recovery, pending replay-plan cleanup, and cached-idle refresh-failure regression coverage
1.11	2026-06-12	BionicLoop engineering	Added trace mapping for generalized issued-dose attribution and different/new-pod no-replay disposition
1.12	2026-06-15	BionicLoop engineering	Updated issued-dose different/new-pod trace mapping to assumed-delivered attribution, nonblocking replacement-pod dosing, and deferred adaptation-forget question
1.13	2026-06-17	BionicLoop engineering	Added trace notes for one-step algorithm/runtime cadence reconciliation before meal-step selection and live executed-step/request-step alignment with algorithm <code>stepTime</code> .

Version	Date	Author	Summary of Changes
1.14	2026-06-18	BionicLoop engineering	Added trace notes for fallback-only replay request-step attribution and evidence-gated issued-dose attribution clearing.
1.15	2026-06-18	BionicLoop engineering	Clarified resolved issued-dose attribution when the dose precedes fallback-active replay: merge into the first fallback replay row rather than replaying pre-fallback disconnected slots.
1.16	2026-06-18	BionicLoop engineering	Added trace notes for retired/expired/no-active-pod fallback recovery using assumed modeled fallback exposure when pump-counter reconciliation evidence is unrecoverable.
1.17	2026-06-23	BionicLoop engineering	Added provisional pod-simulation scaffold trace note and TV-SIM-POD-* working coverage expectations.
1.18	2026-06-23	BionicLoop engineering	Added first pod-simulation scenario slice for shared-invariant same-pod live attribution, replacement-pod assumed delivery, and schedule-weighted fallback replay coverage.
1.19	2026-06-23	BionicLoop engineering	Expanded pod-simulation trace note for canceled and consecutive-canceled meal live attribution, user-escaped unavailable-pod assumed delivery, fallback-maintenance command blocking, and relaunch fallback replay with unavailable CGM.
1.20	2026-06-23	BionicLoop engineering	Added explicit pod-simulation trace coverage for unresolved correction-only and basal-only assumed-delivered replacement-pod dosing.
1.21	2026-06-23	BionicLoop engineering	Added pod-simulation trace coverage for ambiguous fallback restore without replay or modeled-dose injection.
1.22	2026-06-23	BionicLoop engineering	Added pod-simulation trace coverage for assumed old-pod fallback replay, missing-status/nonreplayable fallback-plan clearing, and matching meal-progress cleanup after issued-dose evidence consumption.
1.23	2026-06-23	BionicLoop engineering	Added pod-simulation trace coverage for reservoir-capped partial meal delivery feeding actual delivered units without replay or unresolved meal-progress state.
1.24	2026-06-23	BionicLoop engineering	Added pod-simulation trace coverage for reservoir-capped fallback replay using observed pump delta instead of modeled exposure.
1.25	2026-06-23	BionicLoop engineering	Added pod-simulation trace coverage for meal/fallback overlap partition replay with the meal dose merged into the first fallback-active replay row.
1.26	2026-06-23	BionicLoop engineering	Added trace coverage for app recent-dose persistence/display of merged meal/fallback replay evidence source, disposition, request step, requested units, and delivered units.

Version	Date	Author	Summary of Changes
1.27	2026-06-23	BionicLoop engineering	Added trace coverage for local step CSV export of merged fallback replay evidence source, disposition, and failure-reason fields.
1.28	2026-06-23	BionicLoop engineering	Added trace coverage for runtime cloud step-event emission of merged fallback replay evidence source, disposition, request-step, requested-unit, and delivered-unit fields.
1.29	2026-07-07	BionicLoop engineering	Added trace notes for same-pod fallback recovery using persisted-schedule modeled exposure as assumed delivered when pump delivery delta is unavailable, schedule evidence matches, and no issued-dose partition is required.
1.30	2026-07-08	BionicLoop engineering	Added trace notes for fresh-status patience before same-pod assumed meal resolution after relaunch, algorithm-visible meal-exclusion request-step keying, and cloud/dashboard-only meal-exclusion disposal evidence.
1.31	2026-07-08	BionicLoop engineering	Added trace links for Algo2015 checkBG fingerstick request surfacing, dedupe, haptic feedback, BG-button emphasis, and stale-relaunch suppression.
1.32	2026-07-08	BionicLoop engineering	Added trace links for Algo2015 forced-open-loop characterization, command blocking, fallback-maintenance preservation, and safety-critical alert surfacing.
1.33	2026-07-08	BionicLoop engineering	Added trace links for Reference-host no-boundary-cancel liveness-bound proof across DASH bolus completion, disconnect/unknown state, and fresh idle issued-dose consumption.
1.34	2026-07-09	BionicLoop engineering	Formal freeze-candidate evidence recorded at f098a8b: 14 rows carry executed formal bundles (STR-ALG/AUTO/SIM/SEC lanes) pending sponsor designation; RA-007 hardware closure, RA-015 real-device gap, and RA-009 acceptability disposition remained open.
1.35	2026-07-10	BionicLoop engineering	Reopened the baseline after correcting shared algorithm state: f098a8b marked superseded across all formal-evidence row statuses; added RA-018 (connected-outage zero-insulin window, ANOM-003) and SRS-ALG-008 tracing through RA-014; linked SRS-RUN-007 into RA-003; added the explicit SRS-SEC-003..009 deferral note.
1.36	2026-07-14	BionicLoop engineering	Updated RA-018 for implemented fingerstick-supported CGM-outage dosing: SRS-OUTAGE-001..006 + TV-OUTAGE-001..003 traced with working evidence and formal promotion pending.
1.37	2026-07-14	BionicLoop engineering	RA-013 row gains SRS-CLIN-013/014 and TV-CLIN-014/015 (clinical-feedback defaults + New Participant Reset).

Version	Date	Author	Summary of Changes
1.38	2026-07-14	BionicLoop engineering	Added SRS-PUMP-011 + TV-PUMP-010 to RA-017, corrected its evidence wording to the implemented assumed-delivered semantics, added SDD-OUTAGE-001 to RA-018, extended RA-013 defaults/reset evidence, and added RA-019 for unintended New Participant Reset.
1.39	2026-07-15	BionicLoop engineering	Added SRS-LOG-010 + TV-LOG-010 to RA-008 for build-designated cloud-log integration-evidence collection.
1.40	2026-07-16	BionicLoop engineering	RA-008 row gains SRS-LOG-011 + TV-LOG-011 (server-issued remote logging config).
1.41	2026-07-16	BionicLoop engineering	RA-010 row gains SRS-UI-009 + SDD-APP-008 + TV-UI-011 (server-configurable investigational badge).
1.42	2026-07-18	BionicLoop engineering	Added the RA-020 risk-control row: SRS-OUTAGE-006 / SDD-OUTAGE-001 / TV-OUTAGE-002 with issued-dose crediting and pod-away gap-replay working evidence.
1.43	2026-07-18	BionicLoop engineering	Refreshed RA-020 working evidence for persisted-credit conservation, drift-skip re-synthesis, non-flat quantized weighting, profile-time-zone weighting, and real-C++ engine-ledger conservation. Added the Algo2015 diagnostic-artifact protection regression to RA-009 and corrected its design allocation to SDD-LOG-001.
1.44	2026-07-19	BionicLoop engineering	Exact-step pending manual BG plus meal execution is traced through SRS-BG-004/SRS-MEAL-002, RA-004/RA-012, SDD-BG-001/SDD-POL-004, and TV-MEAL-002. Same-severity Pod signal-loss precedence is traced through SRS-ALERT-011, RA-011, SDD-ALERT-001, and TV-ALERT-002/010.
1.45	2026-07-20	BionicLoop engineering	Manual-BG review safety and direct-request emphasis traced through SRS-BG-001/SRS-ALERT-018/SRS-OUTAGE-003, RA-010/RA-011/RA-012/RA-018, SDD-BG-001/SDD-ALERT-001, and TV-BG-008/TV-ALERT-017/TV-OUTAGE-003/TV-UI-006.
1.46	2026-07-20	BionicLoop engineering	Manual-BG request emphasis rendering hardening traced through SDD-ALERT-001 and TV-ALERT-017; the localized-halo regression test closes the hardware-observed chart compositing artifact without changing request scope, alert precedence, or runtime behavior.
1.47	2026-07-20	BionicLoop engineering	Traced the SRS-PUMP-010 v1.48 fresh-completion upgrade rule through SDD-POL-029, RA-017, and TV-PUMP-009, with adapter restoration bounded by <code>BionicLoopPumpEvidenceRelaunchUpgradeTests</code> following a force-quit mid-bolus field incident.
1.48	2026-07-21	BionicLoop engineering	RA-008 trace note updated for SRS-LOG-009 v1.51 / SDD-LOG-001 v1.52 / TV-LOG-009 v1.47: steady-cadence <code>schedule_checked_unchanged</code> schedule + coverage payload

Version	Date	Author	Summary of Changes
			(no bulk 288-rate array), per-six-hour-bucket observed-slot counts on arm/refresh/unchanged events, and Recent Dose Steps outcome-label/per-segment-provenance rows.
1.49	2026-07-21	BionicLoop engineering	Corrected the SRS-PUMP-010 / SDD-POL-029 / RA-017 / TV-PUMP-009 fresh-completion trace: zero bolusNotDelivered after cancel + relaunch is not full-delivery proof; only explicitly persisted in-progress-capped provenance permits a higher settled observation. Added adapter/runtime/persistence regression links and truth-synced the accepted BG centered shadow bloom after physical-device confirmation.
1.50	2026-07-21	BionicLoop engineering	Prospective forced-open meal blocking traced through SRS-MEAL-002, SDD-POL-004, RA-004, and TV-MEAL-002: persisted qualifying-glucose step, 12/13 blind-step boundary, fresh-CGM/exact-BG exceptions, coordinator pre-persistence backstop, no algorithm/pump activity, and explicit no-delivery copy.
1.51	2026-07-22	BionicLoop engineering	Connected active-delivery meal messaging traced through SRS-MEAL-002, SDD-POL-004, RA-004, and TV-MEAL-002: live .delivering status outranks pending-attribution reconnect copy; idle unresolved attribution remains blocked and unchanged.
1.52	2026-07-22	BionicLoop engineering	Pending-BG open-composer publication-race closure traced through SRS-MEAL-006, SDD-POL-004, RA-004, and TV-MEAL-007: qualifying-glucose publication, not an intermediate executed-step checkpoint, ends the transient wait; qualifying-glucose and pump-status changes trigger revalidation; connected delivery replaces stale backup-basal guidance.
1.53	2026-07-24	BionicLoop engineering	Pregnancy Temporary Target traced through SRS-CLIN-016/017 and SRS-LOG-013, SDD-CLIN-001/002 and SDD-LOG-001, RA-022, and TV-CLIN-017/018 plus TV-LOG-013, with app/core/UI/Scout working evidence and physical/live/formal promotion explicitly pending.
1.54	2026-07-24	BionicLoop engineering	Traced New Participant Reset concurrency and full-wipe closure through SRS-CLIN-014, SDD-CLIN-001/002, RA-019, and TV-CLIN-015: transport quiescence/timeout, durable erase failure rollback, live guard recheck, arm exclusion, and complete outgoing Temporary Target removal. Clarified that same-participant/session reset emits Temporary Target lifecycle evidence, while New Participant Reset erases outgoing identity and pending lifecycle state without a new outgoing-participant event.
1.55	2026-07-30	BionicLoop engineering	Extended RA-010/RA-011 trace coverage for the relocated existing Pod suspend/resume control, duplicate Pod Settings suppression, disabled unavailable/transitional states, unchanged

Version	Date	Author	Summary of Changes
			reminder-only behavior, and direct suspend-alert navigation/copy.
1.56	2026-07-30	BionicLoop engineering	Recorded deferred-scope SRS-SEC-009 working evidence for persistent signed-out login recovery across Home, Alert Center, Account & Session, and notification routing without local algorithm-session interruption.
1.57	2026-07-30	BionicLoop engineering	Added Temporary Target draft-loss, selector-state, Settings-order, and unrelated-copy-removal working evidence to the SRS-CLIN-016 / RA-022 / TV-CLIN-017 trace.
1.58	2026-07-30	BionicLoop engineering	Added persistent Home Manual BG/Meal action placement and expanded-alert scrolling evidence to the SRS-UI-002 / SDD-POL-019 / RA-010 / TV-UI-005 trace.
1.59	2026-07-30	BionicLoop engineering	Traced SRS-LOG-012 permanent-rejection evidence through SDD-LOG-001, RA-008/RA-009, and TV-LOG-012: structured failed-permanent state and Scout gap retained, generic participant alert retired and migrated away, Subject ID Conflict remains specific, and 429 remains retryable.
1.60	2026-07-30	BionicLoop engineering	Traced the suspend-ended direct Resume Insulin action and exact 15-minute safety-critical escalation through SRS-CLIN-016, SDD-CLIN-002/SDD-POL-008, RA-011, and TV-CLIN-017/TV-ALERT-009, including duplicate-command suppression, retryable failure, pump-confirmed clearing, relaunch recovery, and fresh escalation notification.
1.61	2026-07-30	BionicLoop engineering	Traced suspended-meal incident closure through SRS-MEAL-002/SRS-PUMP-010/SRS-UI-002, SDD-POL-004/029, RA-004/017, and TV-MEAL-002/TV-PUMP-009, including idle-only preflight, immediate resume revalidation, and blocked-command cache invalidation.
1.62	2026-07-30	BionicLoop engineering	Added RA-023 trace for explicit/persistent G7 replacement acquisition, staleness non-adoption, ten-minute guidance, and durably deduplicated stall/adoption telemetry through SRS-CGM-006/SRS-ALERT-020/SRS-LOG-014, SDD-CGM-002, and TV-CGM-006/TV-ALERT-019/TV-LOG-014.
1.63	2026-07-31	BionicLoop engineering	Updated RA-015/SRS-ALERT-014/SDD-POL-022/TV-ALERT-013 trace for accurate suspension and fallback-recovery interruption diagnosis, stale blocker clearing, and stronger Home alert precedence; extended RA-019/SRS-CLIN-014/SDD-CLIN-001/TV-CLIN-015 reset trace for asynchronous G7 state isolation.
1.64	2026-08-06	BionicLoop engineering	Extended the RA-008 / SRS-LOG-001 / SDD-LOG-001 / TV-LOG-001 trace with Recent Dose Steps fingerstick provenance, exact value display, and concurrent CGM retention working evidence.

Version	Date	Author	Summary of Changes
1.65	2026-08-11	BionicLoop engineering	Added reference-host total-insulin command trace through RA-014, SRS-ALG-009, SDD-ALG-001, and TV-ALG-012 after field telemetry exposed meal-step basal omission and ~I reconciliation rejection.
1.66	2026-08-11	BionicLoop engineering	Aligned RA-013 trace/evidence with the initial-configuration unlock exception and later-edit lock boundary; removed obsolete feature-branch labels from implemented RA-009/020/021 status notes.
1.67	2026-08-13	BionicLoop engineering	Extended the RA-009 / SRS-LOG-012 / SDD-LOG-001 / TV-LOG-012 trace for the field telemetry backlog incident and its update-safe incremental spool, retained-first drain, aggregate chatter-loss evidence, producer deduplication, and passive health controls.
1.68	2026-08-17	BionicLoop engineering	Extended RA-009 / SRS-LOG-001/012 / SDD-LOG-001 / TV-LOG-001/012 trace for complete active-session step evidence and loss-avoidant SQLite/batch telemetry recovery.
1.69	2026-08-20	BionicLoop engineering	Added SRS-BG-013 / TV-BG-013 trace for the persisted pending-to-used fingerstick chart transition, evidence-only fill, timestamp stability, and deduplication.
1.70	2026-08-20	BionicLoop engineering	Added RA-024 / SRS-LOG-015 / SDD-LOG-001 / TV-LOG-015 trace for clinical-gated, epoch-session-scoped, bounded-snapshot legacy Algo2015 recovery export; extended RA-009 export-security trace.
1.71	2026-08-20	BionicLoop engineering	Extended TV-LOG-015 working trace with the stable single-sheet export-flow regression that prevents picker/alert/share presentation overlap.
1.72	2026-08-20	BionicLoop engineering	Replaced the failed Settings presentation trace with a combined CSV/Algo2015 recovery ZIP delivered through the existing Recent Dose Steps <code>ShareLink</code> , including exact-session and CSV-only mismatch controls.
1.73	2026-08-20	BionicLoop engineering	Extended RA-015 / SRS-ALERT-014 / SDD-POL-022 / TV-ALERT-013 trace for status-refresh-only transition from stale pump-unavailable copy to proven idle-Pod recovery guidance.
1.74	2026-08-20	BionicLoop engineering	Extended RA-010 / SRS-UI-008 / SDD-POL-019 / TV-UI-010 trace for discrete point-only CGM rendering without connector stroke or connected area fill.
1.75	2026-08-21	BionicLoop engineering	Recorded exact-tag formal execution for STR-ALG-001, STR-AUTO-001, STR-SIM-001, and STR-SEC-001; added the authoritative freeze-execution overlay and retained the historical row narratives pending sponsor disposition.

Version	Date	Author	Summary of Changes
1.76	2026-08-21	BionicLoop engineering	Replaced the conflicting exact-freeze overlay plus superseded July matrix with one canonical current 24-row matrix. Exact-freeze evidence and open dispositions now appear on every row; historical candidate states remain in controlled document history.
1.77	2026-08-21	BionicLoop engineering	Added the later controlled CryptoSwift 1.10 source candidate and its working dependency/crypto/core/app/security verification to the RA-009 supporting evidence posture. Preserved all exact-freeze mappings and kept CYBER-DEV-001, formal-evidence disposition, archive identity, and sponsor approval open.
1.78	2026-08-25	BionicLoop engineering	Recorded Camille Powe, MD's written confirmation of 0.01 U schedule-weighted fallback allocation with total conservation, fallback activation when CGM freshness cannot be established, and the 50...500 lb participant weight range with the 20...230 kg algorithm rail. Clinical policy is closed. No trace link, execution result, or product behavior changed.
1.79	2026-08-25	BionicLoop engineering	Added controlled companion evidence closing ALG-DEV-SA-006, AUTO-DEV-001, and AUTO-DEV-002. The exact-freeze static-analysis/linkage companion passed 6/6; the corrected serial UI companion passed 44/44. Original execution records remain unchanged. No requirement, product behavior, hazard control, or risk score changed.
1.80	2026-08-27	BionicLoop engineering	Recorded Camille Powe, MD's approval of seven software/IDE statements: first-step Dexcom requirement, non-synthesized CGM-gap handling, investigational Clinical Settings access, supportive Scout scope with official study sources, no separate formal hardware-validation claim, no separate formal participant usability-study claim, and protocol correction for the distinct fingerstick mode. Updated current dispositions without changing requirements, design, tests, formal results, hazards, controls, or risk scores.
1.81	2026-08-27	BionicLoop engineering	Replaced internal work labels, test-device shorthand, and informal evaluation terminology with direct descriptions of the traced controls and evidence. No trace link, execution result, or product behavior changed.
1.82	2026-08-28	BionicLoop engineering	Added SRS requirement-family child-clause traceability rules and explicit deferred/retired requirement dispositions, including the previously implicit SRS-SEC-004 . . 008 entries. No requirement, design, test, evidence, risk, or product behavior changed.
1.83	2026-08-28	BionicLoop engineering	Added explicit dispositions for all supporting SDD elements and deferred security verification IDs, closing documentation-only orphan gaps found during the technical-appendix review. No

Version	Date	Author	Summary of Changes
			requirement, design, test, evidence, risk, or product behavior changed.
1.84	2026-08-28	Software Developer	Added explicit current-state mappings for TV-SIM-POD-001..004, which passed in the exact-freeze simulation lane but were previously present only through a historical wildcard reference. No test, evidence, risk, or product behavior changed.
1.85	2026-09-03	Software Developer	Added the post-freeze RA-023 adversarial-review disposition to the current trace: exact-freeze app-integration evidence remains passed, while CGM-screen setup reentry, concurrent candidates, automatic-trigger isolation, and first-reading use remain unverified under D06. No requirement, test result, product behavior, hazard score, or formal evidence classification changed.
1.86	2026-09-03	Software Developer	Aligned Build 843 and CYBER-DEV-001 status with the final designation, removed retired/deferred controls from active risk rows, and replaced internal work labels in the current trace. No requirement, test result, product behavior, or hazard score changed.

Current Exact-Freeze Evidence Boundary

This matrix is the canonical current traceability view for the immutable 2026-08-21 engineering freeze. All 24 IDE-scope risk-analysis rows have exact-freeze execution evidence mapped. Execution completion does not constitute sponsor-designated organizational approval, residual-risk acceptance, or approval of a physical-device claim. Working evidence remains support-only unless accepted through the sponsor-approved evidence process.

Primary exact-freeze evidence:

- Evidence/Formal/STR-ALG-001/2026-08-21-ide-freeze-alg-baseline-r2/
- Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/
- Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/
- Archive-matching CryptoSwift companion: Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-cryptoswift-1.9-companion/
- Build 843 controlled CryptoSwift 1.10 verification:
Cybersecurity_CryptoSwift_1_10_Candidate_Verification.md

Superseded candidate states and former working-evidence narratives remain in controlled document and source-control history; they are not current status.

SRS Requirement-Control Traceability

The SRS uses stable parent IDs. For a compound requirement family, each normative shall or shall not occurrence is addressed as the parent ID plus its two-digit textual sequence (for example, SRS-PUMP-010.01). Every child clause inherits the parent family's SDD, RA, TV/STP, and evidence links in the canonical matrix. A parent-family disposition applies only when every applicable child clause has objective evidence or an explicit documented disposition; a passing subset cannot close the parent.

Deferred and Retired Requirements

These identifiers are intentionally outside the frozen-baseline execution claim. They remain represented here so every SRS identifier has a traceable applicability disposition.

Require ment	Applicability	Design / verification disposition	Current disposition
SRS-BG-008	Deferred	SDD-BG-001; TV-BG-007	Step-0 fingerstick rescue is excluded; current step-0 behavior remains governed by SRS-BG-012. Scope re-entry requires SRS/SDD/RA/TV review.
SRS-LOG-010	Retired	Historical SDD-LOG-001; no active verification claim	Build-baked cloud-log profile was removed before participant release. Identifier is retained and shall not be reused.
SRS-SEC-003	Deferred	SDD-AUTH-001; TV-SEC-003	Protected cloud API authentication is outside the claimed package; supportive Scout use is governed by SRS-SEC-001.
SRS-SEC-004	Deferred	SDD-AUTH-001; TV-SEC-004	Multi-provider onboarding policy is outside the claimed package.
SRS-SEC-005	Deferred	SDD-AUTH-001; TV-SEC-005	Cloud role/scope authorization enforcement is outside the claimed package.
SRS-SEC-006	Deferred	SDD-AUTH-001, SDD-POL-015; TV-SEC-003..005	Protected-cloud authentication/session failure behavior is outside the claimed package.
SRS-SEC-007	Deferred	SDD-AUTH-002; TV-SEC-006	Cognito password recovery is outside the claimed package; implementation or working evidence does not promote it.
SRS-SEC-008	Deferred	SDD-AUTH-001, SDD-POL-016; TV-SEC-007	Authenticated launch session restoration is outside the claimed package; implementation or working evidence does not promote it.
SRS-SEC-009	Deferred	SDD-POL-015, SDD-POL-016;	Signed-out local Home bypass/login recovery is outside the claimed package; existing working evidence remains support-

Requirement	Applicability	Design / verification disposition	Current disposition
		TV-SEC-008	only.

Supporting Design-Element Disposition

The canonical risk rows list design controls most directly tied to each hazard. The following active supporting elements are not repeated on individual risk rows but remain traced to the requirement families shown. Deferred authentication elements remain implementation context and do not create a closure claim.

Design element	Applicability	Trace disposition
SDD-APP-002	Active	Runtime persistence support for SRS-RUN-002 and SRS-STATE-001..005
SDD-APP-004	Active	Pump signal-loss and alert-state support for SRS-PUMP-004 and SRS-ALERT-001..011
SDD-APP-005	Active	App-layer telemetry support for SRS-LOG-001..015
SDD-APP-006	Active	Work-execution sequencing support for SRS-RUN-001..003
SDD-CFG-001	Active	Configuration/dependency control supporting SRS-SEC-001..002 and the Cybersecurity Plan
SDD-CORE-001	Active	Core due-step/input/algorithm/command orchestration supporting runtime, BG, meal, pump, and state requirements
SDD-SIM-001	Verification support	Simulation architecture supporting TV-SIM-001..005; not a production risk control
SDD-AUTH-001	Deferred context	Implementation context for SRS-SEC-003..006 and SRS-SEC-008; no current closure claim
SDD-AUTH-002	Deferred context	Implementation context for SRS-SEC-007; no current closure claim
SDD-POL-015	Deferred context	Authentication-failure policy context for SRS-SEC-006 and SRS-SEC-009; no current closure claim
SDD-POL-016	Deferred context	Session-restore policy context for SRS-SEC-008..009; no current closure claim

Supporting and Deferred Verification-Element Disposition

The exact-freeze simulation lane passed the four pod-simulation verification elements below. They supplement the verification links in the canonical risk rows and share the controlled STR-SIM-001 evidence locator.

Verification element	Applicability	Trace disposition
TV-SIM-POD-001	Active verification support	Stateful pod-ledger, mask, expiry, and replacement-identity coverage supporting RA-003, RA-015, and RA-017
TV-SIM-POD-002	Active verification support	Persisted issued-dose restoration and exactly-once replay coverage supporting RA-017
TV-SIM-POD-003	Active verification support	Meal/correction and fallback-total partitioning coverage supporting RA-017 and RA-020
TV-SIM-POD-004	Active verification support	High-risk relaunch, replacement-pod, fallback, meal, and evidence-preservation scenarios supporting RA-003, RA-004, RA-017, and RA-020

TV-SEC-002 is deferred because Scout is approved as a supportive service, not an official outcome source or local dosing dependency. It remains the predefined verification method if that boundary changes. TV-SEC-003 . . 008 are dispositioned with their deferred requirements in the table above. No deferred verification ID is represented as executed or accepted evidence.

Canonical IDE-Scope Matrix

RA-001

- **Requirements:** SRS-RUN-001, SRS-RUN-002, SRS-RUN-003
- **Design:** SDD-POL-001, SDD-APP-003
- **Verification:** TV-RUN-001, TV-RUN-002, TV-RUN-003, TV-SIM-001
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM pass
- **Remaining disposition:** Sponsor trace acceptance/signature

RA-002

- **Requirements:** SRS-CGM-001, SRS-CGM-002, SRS-CGM-003, SRS-CGM-004
- **Design:** SDD-POL-002, SDD-CGM-001

- **Verification:** TV-CGM-001, TV-CGM-002, TV-CGM-003, TV-CGM-004, TV-SIM-002
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM pass
- **Remaining disposition:** Separate formal G7 hardware-validation claim not retained; sponsor trace acceptance remains

RA-003

- **Requirements:** SRS-PUMP-001, SRS-PUMP-002, SRS-PUMP-005, SRS-PUMP-006, SRS-PUMP-007, SRS-PUMP-008, SRS-PUMP-009, SRS-RUN-006, SRS-RUN-007, SRS-STATE-005
- **Design:** SDD-POL-003, SDD-POL-028, SDD-PUMP-001, SDD-DATA-003
- **Verification:** TV-PUMP-001, TV-PUMP-002, TV-PUMP-005, TV-PUMP-006, TV-PUMP-007, TV-PUMP-008, TV-STATE-005, TV-RUN-008, TV-SIM-003
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Separate formal hardware-validation claim not retained; sponsor trace acceptance remains

RA-004

- **Requirements:** SRS-MEAL-001, SRS-MEAL-002, SRS-MEAL-003, SRS-MEAL-004, SRS-MEAL-005, SRS-MEAL-006
- **Design:** SDD-POL-004, SDD-APP-001
- **Verification:** TV-MEAL-001, TV-MEAL-002, TV-MEAL-003, TV-MEAL-004, TV-MEAL-005, TV-MEAL-006, TV-MEAL-007, TV-SIM-004
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Separate formal physical meal claim not retained; sponsor trace acceptance remains

RA-005

- **Requirements:** SRS-PUMP-003, SRS-LOG-001
- **Design:** SDD-PUMP-001, SDD-LOG-001
- **Verification:** TV-PUMP-003, TV-LOG-001
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/

- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Sponsor trace acceptance/signature

RA-006

- **Requirements:** SRS-STATE-001, SRS-STATE-002, SRS-STATE-003, SRS-STATE-004, SRS-STATE-005
- **Design:** SDD-DATA-001, SDD-DATA-002, SDD-DATA-003, SDD-DATA-004, SDD-POL-005, SDD-POL-028, SDD-PUMP-001
- **Verification:** TV-STATE-001, TV-STATE-002, TV-STATE-003, TV-STATE-004, TV-STATE-005
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Separate formal physical relaunch/reset claim not retained; sponsor trace acceptance remains

RA-007

- **Requirements:** SRS-PUMP-004, SRS-PUMP-005
- **Design:** SDD-PUMP-001
- **Verification:** TV-PUMP-004, TV-PUMP-005
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/
- **Execution state:** Automated support exists; no hardware STR executed
- **Remaining disposition:** STP-HW-001 not retained as a separate formal claim; sponsor trace acceptance remains

RA-008

- **Requirements:** SRS-LOG-001, SRS-LOG-002, SRS-LOG-003, SRS-LOG-004, SRS-LOG-005, SRS-LOG-006, SRS-LOG-007, SRS-LOG-008, SRS-LOG-009, SRS-LOG-012
- **Design:** SDD-LOG-001, SDD-POL-017, SDD-POL-018, SDD-POL-024, SDD-POL-025, SDD-APP-007, SDD-DATA-003
- **Verification:** TV-LOG-001, TV-LOG-002, TV-LOG-003, TV-LOG-004, TV-LOG-005, TV-LOG-006, TV-LOG-007, TV-LOG-008, TV-LOG-009, TV-LOG-012
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/
- **Execution state:** AUTO unit pass + SEC local-control pass
- **Remaining disposition:** Live retained-sequence observation if retained

RA-009

- **Requirements:** SRS-SEC-001, SRS-SEC-002, SRS-LOG-012, SRS-LOG-015

- **Design:** SDD-LOG-001 + CybersecurityPlan.md
- **Verification:** TV-SEC-001, TV-LOG-012, TV-LOG-015
- **Exact-freeze evidence:** Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/; Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-cryptoswift-1.9-companion/
- **Execution state:** SEC local-control pass (68/68)
- **Disposition:** Verification evidence and traceability accepted under D08; cybersecurity scope and controls accepted under D09. Scout/live-sequence and physical ZIP observations remain corroborating/supporting evidence.

RA-010

- **Requirements:** SRS-UI-001, SRS-UI-002, SRS-UI-003, SRS-UI-004, SRS-UI-005, SRS-UI-006, SRS-UI-007, SRS-UI-008, SRS-UI-009, SRS-CLIN-016, SRS-VAL-001, SRS-BG-001
- **Design:** SDD-POL-006, SDD-POL-007, SDD-POL-009, SDD-POL-014, SDD-POL-018, SDD-POL-019, SDD-APP-008, SDD-CLIN-002
- **Verification:** TV-UI-001, TV-UI-002, TV-UI-003, TV-UI-004, TV-UI-005, TV-UI-006, TV-UI-007, TV-UI-008, TV-UI-009, TV-UI-010, TV-UI-011, TV-CLIN-017
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-AUTO-001/2026-08-25-ui-harness-companion/
- **Execution state:** AUTO unit pass; corrected serial UI companion 44/44; AUTO-DEV-002 closed
- **Remaining disposition:** Separate formal physical visual claim not retained; sponsor trace acceptance remains

RA-011

- **Requirements:** SRS-ALERT-001, SRS-ALERT-002, SRS-ALERT-003, SRS-ALERT-004, SRS-ALERT-005, SRS-ALERT-006, SRS-ALERT-007, SRS-ALERT-008, SRS-ALERT-009, SRS-ALERT-010, SRS-ALERT-011, SRS-ALERT-012, SRS-ALERT-015, SRS-ALERT-016, SRS-ALERT-017, SRS-ALERT-018, SRS-ALERT-019
- **Design:** SDD-ALERT-001, SDD-POL-008, SDD-POL-026, SDD-DATA-005
- **Verification:** TV-ALERT-001, TV-ALERT-002, TV-ALERT-003, TV-ALERT-004, TV-ALERT-005, TV-ALERT-006, TV-ALERT-007, TV-ALERT-008, TV-ALERT-009, TV-ALERT-010, TV-ALERT-011, TV-ALERT-014, TV-ALERT-015, TV-ALERT-016, TV-ALERT-017, TV-ALERT-018, TV-SIM-005
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM alert pass
- **Remaining disposition:** Supplemental alert drill only if retained

RA-012

- **Requirements:** SRS-BG-001, SRS-BG-002, SRS-BG-003, SRS-BG-004, SRS-BG-005, SRS-BG-006, SRS-BG-007, SRS-BG-009, SRS-BG-010, SRS-BG-011, SRS-BG-012, SRS-BG-013
- **Design:** SDD-BG-001, SDD-POL-010, SDD-POL-011, SDD-POL-012, SDD-LOG-001
- **Verification:** TV-BG-001, TV-BG-002, TV-BG-003, TV-BG-004, TV-BG-005, TV-BG-006, TV-BG-008, TV-BG-009, TV-BG-010, TV-BG-011, TV-BG-012, TV-BG-013
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM pass
- **Remaining disposition:** Sponsor trace acceptance/signature

RA-013

- **Requirements:** SRS-CLIN-001, SRS-CLIN-002, SRS-CLIN-003, SRS-CLIN-004, SRS-CLIN-005, SRS-CLIN-006, SRS-CLIN-007, SRS-CLIN-008, SRS-CLIN-009, SRS-CLIN-010, SRS-CLIN-011, SRS-CLIN-012, SRS-CLIN-013, SRS-CLIN-014, SRS-VAL-001, SRS-LOG-008
- **Design:** SDD-CLIN-001, SDD-POL-013, SDD-POL-017, SDD-POL-025, SDD-DATA-006
- **Verification:** TV-CLIN-001, TV-CLIN-002, TV-CLIN-003, TV-CLIN-004, TV-CLIN-005, TV-CLIN-006, TV-CLIN-007, TV-CLIN-008, TV-CLIN-009, TV-CLIN-010, TV-CLIN-011, TV-CLIN-012, TV-CLIN-013, TV-CLIN-014, TV-CLIN-015, TV-LOG-008
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-AUTO-001/2026-08-25-ui-harness-companion/
- **Execution state:** AUTO unit pass; corrected serial UI companion 44/44; AUTO-DEV-001 closed
- **Remaining disposition:** Unlock operations record

RA-014

- **Requirements:** SRS-ALG-001, SRS-ALG-002, SRS-ALG-003, SRS-ALG-004, SRS-ALG-005, SRS-ALG-006, SRS-ALG-007, SRS-ALG-008, SRS-ALG-009
- **Design:** SDD-ALG-001, SDD-QA-001
- **Verification:** TV-ALG-001, TV-ALG-002, TV-ALG-003, TV-ALG-004, TV-ALG-005, TV-ALG-006, TV-ALG-007, TV-ALG-008, TV-ALG-009, TV-ALG-010, TV-ALG-011, TV-ALG-012
- **Exact-freeze evidence:** Evidence/Formal/STR-ALG-001/2026-08-21-ide-freeze-alg-baseline-r2/; Evidence/Formal/STR-ALG-001/2026-08-25-sa-006-linkage-companion/
- **Execution state:** ALG behavioral pass; exact-freeze static-analysis/linkage companion 6/6; ALG-DEV-SA-006 closed
- **Remaining disposition:** None for ALG-DEV-SA-006

RA-015

- **Requirements:** SRS-CGM-005, SRS-RUN-004, SRS-RUN-005, SRS-ALERT-013, SRS-ALERT-014, SRS-UI-001, SRS-UI-002
- **Design:** SDD-POL-020, SDD-POL-021, SDD-POL-022, SDD-POL-008, SDD-APP-003, SDD-CGM-001
- **Verification:** TV-CGM-005, TV-RUN-004, TV-RUN-005, TV-RUN-006, TV-RUN-007, TV-ALERT-012, TV-ALERT-013, TV-UI-001
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM pass
- **Remaining disposition:** Separate formal hardware-recovery claim not retained; sponsor trace acceptance remains

RA-016

- **Requirements:** SRS-MEAL-007, SRS-MEAL-008, SRS-MEAL-009, SRS-MEAL-010, SRS-MEAL-011, SRS-MEAL-012, SRS-LOG-007, SRS-UI-002
- **Design:** SDD-POL-023, SDD-POL-024, SDD-POL-027, SDD-POL-029, SDD-APP-001, SDD-LOG-001
- **Verification:** TV-MEAL-008, TV-MEAL-009, TV-MEAL-010, TV-MEAL-011, TV-MEAL-012, TV-MEAL-013, TV-LOG-007, TV-PUMP-003
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Separate formal physical meal-lifecycle claim not retained; sponsor trace acceptance remains

RA-017

- **Requirements:** SRS-PUMP-010, SRS-PUMP-011, SRS-MEAL-012, SRS-STATE-004, SRS-STATE-005, SRS-LOG-001
- **Design:** SDD-POL-029, SDD-PUMP-001, SDD-DATA-003, SDD-LOG-001
- **Verification:** TV-PUMP-009, TV-PUMP-010, TV-MEAL-013, TV-STATE-004, TV-STATE-005, TV-LOG-001
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/
- **Execution state:** AUTO unit pass + SIM/pod-sim pass
- **Remaining disposition:** Separate formal physical canceled-partial claim not retained; sponsor residual-risk/trace acceptance remains

RA-018

- **Requirements:** SRS-ALERT-019, SRS-RUN-007, SRS-ALG-008, SRS-OUTAGE-001, SRS-OUTAGE-002, SRS-OUTAGE-003, SRS-OUTAGE-004, SRS-OUTAGE-005, SRS-OUTAGE-006
- **Design:** SDD-ALERT-001, SDD-ALG-001, SDD-PUMP-001, SDD-OUTAGE-001
- **Verification:** TV-ALERT-018, TV-ALG-004, TV-OUTAGE-001, TV-OUTAGE-002, TV-OUTAGE-003
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/; Evidence/Formal/STR-ALG-001/2026-08-21-ide-freeze-alg-baseline-r2/
- **Execution state:** AUTO unit pass + SIM/algo-sim pass
- **Disposition:** Residual risk accepted under D07 and traceability accepted under D08; no separate formal outage hardware claim is retained.

RA-019

- **Requirements:** SRS-CLIN-014
- **Design:** SDD-CLIN-001
- **Verification:** TV-CLIN-015
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/
- **Execution state:** AUTO unit pass + SEC persistence support
- **Remaining disposition:** Sponsor trace acceptance/signature

RA-020

- **Requirements:** SRS-OUTAGE-006
- **Design:** SDD-OUTAGE-001
- **Verification:** TV-OUTAGE-002
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/; Evidence/Formal/STR-ALG-001/2026-08-21-ide-freeze-alg-baseline-r2/
- **Execution state:** AUTO unit pass + SIM/pod-sim/algo-sim pass
- **Remaining disposition:** Clinical quantization policy confirmed; sponsor-designated residual-risk/trace acceptance remains; separate formal hardware claim not retained

RA-021

- **Requirements:** SRS-CLIN-015
- **Design:** SDD-CLIN-001
- **Verification:** TV-CLIN-016

- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/
- **Execution state:** AUTO unit pass
- **Remaining disposition:** Clinical weight range confirmed; sponsor-designated residual-risk/trace acceptance

RA-022

- **Requirements:** SRS-CLIN-016, SRS-CLIN-017, SRS-LOG-013, supporting SRS-RUN-007
- **Design:** SDD-CLIN-002, SDD-CLIN-001, SDD-LOG-001
- **Verification:** TV-CLIN-017, TV-CLIN-018, TV-LOG-013
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/
- **Execution state:** AUTO unit pass; UI target deviation does not involve Temporary Target tests
- **Remaining disposition:** Sponsor trace acceptance; lifecycle telemetry remains corroborating and separate formal physical evidence is not claimed

RA-023

- **Requirements:** SRS-CGM-006, SRS-ALERT-020, SRS-LOG-014
- **Design:** SDD-CGM-002
- **Verification:** TV-CGM-006, TV-ALERT-019, TV-LOG-014
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/
- **Execution state:** AUTO unit pass for the tested app-integration scope; the September 3 adversarial review identified untested implementation paths involving CGM-screen setup reentry, concurrent candidates, automatic-trigger isolation, and first-reading use
- **Disposition:** D06 accepts unchanged Build 843 for submission with the controlled multi-device exercise and final RA-023 disposition required before participant deployment. Separate formal multi-device hardware validation is not claimed, and live telemetry remains corroborating.

RA-024

- **Requirements:** SRS-LOG-015, SRS-SEC-001
- **Design:** SDD-LOG-001
- **Verification:** TV-LOG-015
- **Exact-freeze evidence:** Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/; Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/
- **Execution state:** AUTO unit pass + SEC recovery-export pass
- **Remaining disposition:** Sponsor trace acceptance; physical ZIP observation remains supporting evidence rather than a separate formal claim

Current Interpretation Rules

- STP-ALG-001 passed behaviorally; the exact-freeze 6/6 linkage companion closes ALG-DEV-SA-006.
- STP-AUTO-001 completed 997 app tests: 996 passed, one intentional IFU reference-table generation helper skip gated by an export directory, zero failures. The original UI run recorded 41 passes and three reproducible harness deviations; the corrected serial UI companion passed 44/44 and closes AUTO-DEV-001 and AUTO-DEV-002.
- STP-SIM-001 passed.
- TV-SEC-001 passed 68/68 scoped controls for the exact freeze. Build 843 uses the controlled CryptoSwift 1.10.0 resolution and is the IDE submission build designated on 2026-09-03. CYBER-DEV-001 is closed through the selected "designate and verify a new controlled-resolution build" path; evidence acceptance was recorded under D08 and D09.
- A separate formal hardware-validation study and supplemental physical/alert claims are not retained; real-Pod formative records remain supporting evidence under the 2026-08-27 clinical approval.
- Sponsor residual-risk approval, evidence acceptance, cybersecurity acceptance, and controlled-document approval references are closed under D07 through D09 and D11. The three decisions recorded in `IDE_Clinical_Policy_Confirmation_2026-08-21.md` and the seven decisions in `Evidence/ClinicalPolicy/2026-08-27-software-ide-decisions/README.md` are complete.

Usage

For each controlled change:

1. update affected SRS-*, SDD-*, RA-*, and TV-* links
2. identify the controlled evidence bundle and execution result
3. distinguish working evidence from formally promoted evidence
4. record remaining Sponsor, Clinical/Regulatory, operator, or designated software-evidence approver disposition
5. obtain the required reviewer and approver signatures before changing a row to accepted or closed

Cybersecurity Plan (Investigational Build)

Status: Frozen-baseline cybersecurity scope and controls approved for IDE submission Version:

1.03 Owner: BionicLoop engineering

Prepared by: BionicLoop engineering Approval reference: Edward R. Damiano, PhD, cybersecurity scope/control acceptance and final software-package approval, controlled email dated 2026-09-07 EDT Baseline freeze SHA: 91c0e98a9bc9429a0486bebdebf6c7d8dbbe300e Last updated: 2026-09-08

Revision History

Version	Date	Author	Summary of Changes
0.1	2026-03-27	Engineering	Initial cybersecurity draft baseline
0.9	2026-04-06	BionicLoop engineering	Added controlled-document metadata and expanded the cybersecurity package with trust boundaries, inherited-control mapping, local-control mapping, and explicit artifact gaps.
0.91	2026-07-09	BionicLoop engineering	Added threat scenarios for persisted fallback/issued-dose recovery state and clinical-unlock verifier material; added update/patch-delivery, end-of-support, and vulnerability-cadence rows to open cybersecurity gaps.
0.92	2026-07-10	BionicLoop engineering	Added Algo2015 and Algo2015Safety to the trust-boundary SOUP statement, supplier matrix, SBOM scope, threat scenarios, and planned evidence; recorded separately controlled source custody and build-identity evidence; aligned the investigational export posture with RA-009.
0.93	2026-07-14	BionicLoop engineering	Added the unlock-gated, dual-confirmed New Participant Reset as the between-participant local-data lifecycle control and aligned the related threat scenario.
0.94	2026-07-18	BionicLoop engineering	Documentation truth-sync for the sponsor-approved RA-009 closure: removed superseded claims that Files sharing and an unprotected Documents CSV remain enabled; recorded the gated Application Support export, absent sharing/open-in-place keys, protected and backup-excluded local artifacts, sequenced retained telemetry, and remaining freeze-time verification gates.
0.95	2026-07-20	BionicLoop engineering	Recorded the pre-freeze cybersecurity rehearsal: seven focused TV-SEC-001 regressions passed as working evidence, tracked-source credential-pattern scan found no matches in reviewed scope, both algorithm binaries and current dependency pins/checksums are now inventoried, and formal/live/scanner/supplier closure boundaries remain explicit.

Version	Date	Author	Summary of Changes
0.96	2026-08-21	BionicLoop engineering	Aligned the local-control baseline with update-safe telemetry migration, the uncapped protected clinical ledger, bounded batch recovery, complete active-session step evidence, and the clinical-gated combined CSV/Algo2015 recovery ZIP with epoch-session isolation and protected staging.
0.97	2026-08-21	BionicLoop engineering	Recorded the exact-freeze 68/68 local-control pass and the freeze-time composition/manual-advisory companion; captured current algorithm-binary hashes and safety identity; and opened CYBER-DEV-001 because ignored Package.resolved state allowed archive 826 and the formal run to use CryptoSwift 1.9.0 and 1.10.0 respectively.
0.98	2026-08-21	BionicLoop engineering	Bound the plan to the three-layer FDA package, removed a workstation-specific internal link, and explicitly escalated the conflict between deferred cloud verification and the statement that Scout is a primary study transport. No cybersecurity control or product behavior changed.
0.99	2026-08-21	BionicLoop engineering	Recorded the controlled-source candidate that requires CryptoSwift 1.10.0 exactly, tracks the app-workspace resolution, and passes dependency, crypto/session, core, app, and scoped-security verification. Preserved archive 826 and exact-freeze evidence as immutable history and kept CYBER-DEV-001 open pending a designated archive, physical transport check, evidence disposition, and sponsor approval.
1.00	2026-08-27	BionicLoop engineering	Recorded the approved supportive BionicScout scope and replaced internal review labels with direct descriptions of the affected cybersecurity controls and evidence. No cybersecurity control or product behavior changed.
1.01	2026-08-27	BionicLoop engineering	Added the Build 843 CycloneDX SBOM, dated advisory/license review, and IDE cybersecurity closure matrix; aligned Scout sequence evidence with its approved corroborating role. No product behavior changed.
1.02	2026-08-28	Software Developer	Separated available cybersecurity evidence from still-open additions, removed stale planned-evidence wording, and clarified the historical archive-826 dependency deviation relative to Build 843. No cybersecurity control, test result, or product behavior changed.
1.03	2026-09-03	Software Developer	Recorded the 2026-09-03 Build 843 IDE-submission designation and closure of CYBER-DEV-001, reconciled the completed permission and RTM evidence, and replaced placeholder incident-response tasks with controlled governance references. No cybersecurity control, test result, or product behavior changed.

1. Scope

This plan covers cybersecurity controls for:

- mobile app runtime
- connected-device communications (CGM + pump)
- telemetry data handling (local persistence and cloud upload path)

Current submission scope boundary:

- In the current software package:
 - local/device-resident telemetry handling and export controls
 - software design statements for secure telemetry architecture
 - documentation of implemented auth/session code paths where they affect software understanding
- Deferred from the current software package:
 - formal closure of cloud-upload verification
 - protected API access, onboarding-provider, authorization-role, password-recovery, session-restore, and auth-failure continuity claims represented by SRS-SEC-003..009

2. Security Objectives

- Preserve safety-critical command integrity.
- Protect confidentiality of participant data.
- Maintain availability of safety-relevant status and logs.

3. System Trust Boundary Summary

Primary trust-boundary elements for the current software package:

- iPhone-resident BionicLoop application
- Dexcom G7 sensor communication path and supporting official Dexcom application
- Omnipod DASH pod communication path
- local persistence (UserDefaults, protected Application Support/app-container files, and Keychain)
- local alerting and notification surfaces
- cloud telemetry upload and retained-record delivery path

Current boundary statement:

- BionicLoop owns the controller-app software behavior, local persistence choices, local export behavior, alerting behavior, and cloud-client code it ships.

- Dexcom and Insulet own the cleared device firmware, proprietary radio/device behavior, and primary CGM/pump safety/security controls of their respective products.
- Local package code (OmniBLE, G7SensorKit, LoopKit) is treated as software of unknown provenance / inherited open-source software that requires version, provenance, and delta tracking by BionicLoop.
- The embedded dosing-algorithm binaries (Algo2015.xcframework and the symbol-isolated Algo2015Safety.xcframework) are built locally from the separately controlled Algo2015 C++ reference source through the controlled primary and safety build processes; provenance is anchored by algorithm-source custody plus the safety build's identity report (byte-identical object except exactly three renamed entry symbols).

4. Supplier / Inherited Control Matrix

Component	Security Property Relied Upon	Current Controlled Evidence	Additional Artifact Needed for Submission Closure	BionicLoop Ownership Boundary
OmniPod DASH pod / radio session	Device-side command/session security and encrypted command transport	Local OmniBLE implementation uses LTK-backed encrypted DASH transport with nonce sequencing and AES-CCM handling in OmniBLE/OmniBLE/Bluetooth/Encrypt/Decrypt.swift, OmniBLE/OmniBLE/PumpManager/MessageTransport.swift, and OmniBLE/OmniBLE/PumpManager/PodComms.swift.	Insulet / FDA-cleared product cybersecurity and interoperability artifact or approved supplier summary identifying the relied-upon control.	BionicLoop may rely on the pod/session security model but does not claim original ownership of DASH firmware/radio security.
Dexcom G7 sensor / radio session	Device-side authenticated CGM session behavior	Local G7 integration exposes a dedicated authentication characteristic and authenticated service boundary in G7SensorKit/G7SensorKit/BluetoothServices.swift; the plugin README also requires use of the official G7 app in G7SensorKit/README.md.	Dexcom / FDA-cleared product cybersecurity and interoperability artifact or approved supplier summary identifying the relied-upon control.	BionicLoop may rely on the G7 session model and official Dexcom app requirement but does not claim original ownership of Dexcom device/app security.
Official Dexco	Primary CGM alarm and safety alerting	Current BionicLoop requirements and IFU explicitly state Dexcom remains the source of truth for CGM alarming.	Formal citation package for Dexcom	BionicLoop owns only its supplemental in-

Comp onent	Security Property Relied Upon	Current Controlled Evidence	Additional Artifact Needed for Submission Closure	BionicLoop Ownership Boundary
m applic ation			alarm/security behavior used by the study / submission team.	app CGM status/review presentation.
OmniB LE local packa ge	Pump communicatio n implementatio n	Controlled local source and tests are available; the provenance review records the likely public upstream source, likely import commit basis, shipped local delta summary, and classification of reviewed transport changes as connection/pairing recovery rather than cryptographic weakening.	Any available upstream tag/release mapping for the identified import basis.	BionicLoop owns the chosen version and any local modifications it ships.
G7Sen sorKit local packa ge	CGM communicatio n implementatio n	Controlled local source is available; the package README documents the official G7 app dependency; the provenance review records the likely public upstream source, likely initial import and later synchronization basis, shipped local delta summary, and no security-semantic local delta in the reviewed BLE/auth/logging surfaces.	Any available upstream tag/release mapping for the identified import/sync basis.	BionicLoop owns the chosen version and any local modifications it ships.
LoopKit local packa ge	Shared data storage / platform helper library behavior	Controlled local source and license are available; the provenance review records the likely public upstream source, likely import commit basis, shipped local delta summary, and no security-semantic local delta in the reviewed auth/keychain/logging surfaces.	Any available upstream tag/release mapping for the identified import basis.	BionicLoop owns the chosen version and any local modifications it ships.
Algo2015 / Algo2015Safety binary targets	Dosing- algorithm computation integrity (unmodified reference algorithm; per- instance state isolation for the safety track)	The controlled primary and safety build processes produce a safety-build identity report proving byte identity to the primary compilation except three renamed entry symbols, with all other globals localized; isolation is regression-tested (Algo2015DualInstanceIsolationTests). Algorithm C++ source is controlled separately under a no-modification rule.	Freeze-time SBOM/dependency-inventory rows for both binaries plus the archived identity report at the freeze SHA.	BionicLoop owns the hosting, build, and isolation of the binaries; the algorithm's clinical logic is the sponsor-controlled reference source and is never modified.

5. BionicLoop-Owned Local Security Controls

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
Token and credential storage	Auth tokens and stored credentials are persisted in iOS Keychain with <code>kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly</code> .	<code>BionicLoop/App/AuthSessionNetworking.swift</code>	Applies only to auth/session data, not to all local telemetry artifacts.
Authenticated API requests	Protected requests carry Authorization: Bearer <token> and retry once after 401 using refreshed token.	<code>BionicLoop/App/AuthenticatedAPIClient.swift</code> , <code>BionicLoop/App/AuthSessionNetworking.swift</code>	Formal closure for protected cloud/API scope remains deferred in this software package.
Local telemetry outbox persistence	Telemetry uses update-safe incremental migration into a protected SQLite ledger. Retained clinical records are not count-capped, receive monotonic sequence	<code>BionicLoop/App/CloudTelemetryReporter.swift</code> , <code>BionicLoop/App/CloudTelemetryOutbox.swift</code> , <code>BionicLoop/App/SQLiteCloudTelemetryOutboxLedger.swift</code>	Exact-freeze local-control evidence is complete. Available sequence-correlated phone-to-Scout observations are corroborating operational evidence; local capacity remains bounded by device storage and iOS app-container availability.

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
	numbers, and drain through bounded batch requests with backward-compatible single-record fallback; retryable and permanent failures remain durable.		
Between - participant local-data lifecycle	New Participant Reset (2026-07-14) wipes every participant-scoped store on a reused study phone - clinical configuration, local step telemetry and CSV export, cloud outbox queue, alert history, stored pump/CGM manager state, subject-scoped Keychain unlock material including burned	BionicLoop/Runtime/LoopRuntimeEngineNewParticipantResetsupport.swift	Supports the RA-009 local-data posture between participants; formal promotion follows the clinical-settings lane

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
	counters, onboarding claim - then signs out; unlock-gated, dual-confirmed, refused with an active pod (RA-013/RA-019, SRS-CLIN-014, TV-CLIN-015).		
Export minimization boundary	A bounded complete active-session step CSV is snapshotted into protected staging and can leave the app only through the clinical-gated Recent Dose Steps share flow. The same ZIP may include byte-faithful copies of only the matching/current epoch-session Algo2015 artifacts; missing or mismatched artifacts yield disclosed	BionicLoop/Runtime/LoopTelemetryStore.swift, BionicLoop/App/LegacyAlgorithmArtifactExport.swift, BionicLoop/Features/Home/HomeRecentDoseStepsView.swift, BionicLoop/Resources/Info.plist	Exact-freeze TV-SEC-001 and Build 843 recovery-ZIP observations cover protection, backup exclusion, session isolation, gated sharing, ZIP contents, and absence of sharing keys; evidence acceptance is recorded under D08 and D09.

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
	partial or CSV-only output, never cross-session mixing. No participant-build Files sharing/open-in-place path exposes the source files.		
File-access surface	The sponsor-approved RA-009 closure removes UIFileSharingEnabled / LSSupportsOpeningDocumentsInPlace; the step archive, telemetry ledger, and staged recovery ZIP use protected, backup-excluded app-container storage. Algo2015 inspection artifacts remain in the app container because the C++ engine	BionicLoop/Resources/Info.plist, LoopTelemetryStore.swift, CloudTelemetryOutbox.swift, LegacyAlgorithmArtifactExport.swift, Algo2015ArtifactTailSupport.swift	Exact-freeze local-control evidence covers the app-owned paths. Protection remains best-effort for C++-created live artifacts; no ungated external file surface is claimed.

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
	owns their location, but the app applies protection and backup exclusion when it enumerates or snapshots them.		
Retained telemetry delivery integrity	Safety-relevant retained records use the durable protected ledger and monotonic retained-record sequence numbers so the server can detect loss without normal diagnostic chatter creating false gaps. A separate protected complete active-session step archive preserves local clinical step evidence independently from the bounded	BionicLoop/App/SQLiteCloudTelemetryOutboxLedger.swift, BionicLoop/App/CloudTelemetryReporter.swift, BionicLoop/Runtime/LoopTelemetryStore.swift, cloud telemetry contract tests	Local freeze evidence is complete. Sequence-correlated Scout data is corroborating and supports material-gap investigation; it is not required to establish local dosing/session reconstruction. Storage-pressure operational disposition remains open.

Control Area	Current Implementation / Observation	Controlled Evidence	Current Limitation / Residual Gap
	Home display cache.		
Logging discipline	Debug/API/telemetry logging redacts request query strings and avoids dumping bearer tokens directly.	BionicLoop/App/AuthenticatedAPIClient.swift, BionicLoop/App/CloudTelemetryReporter.swift, BionicLoop/App/AuthSessionNetworking.swift, , Docs/Quality/Cybersecurity_Logging_and_Secret_Review.md	Reviewed for the current software package, but debug-response-snippet behavior and console observability remain a residual development-risk consideration if auth/cloud scope is later reopened.
Platform permissions	Current app declares Bluetooth-central background use and Bluetooth usage descriptions.	BionicLoop/Resources/Info.plist; Cybersecurity_Local_File_and_Permission_Review.md	The controlled least-privilege review is complete for Build 843. Any future permission, background mode, or entitlement change requires a new review.

6. Security Baseline Controls

- Transport security for all cloud telemetry endpoints.
- Minimize external exposure of persistent local sensitive data; keep retained telemetry, the active-session step archive, and recovery staging protected and backup-excluded, and expose a bounded recovery ZIP solely through the clinical-gated in-app share flow.
- Principle of least privilege for iOS permissions and background modes; participant builds omit Files sharing and open-in-place capabilities.
- Dependency and package version tracking (SBOM-ready inventory), including the locally built Algo2015 / Algo2015Safety binary targets.
- Logging controls: avoid secrets and unnecessary identifiers in clear text.
- Explicit failure handling for unavailable/unknown device states.

- Identity and access control implementation exists in development builds, but formal closure for protected auth/provider/role flows is deferred from the current software package.

7. Threat Scenarios (Current Focus)

- BLE disruption or spoofed/disrupted communication causing stale status.
- Unauthorized use of the clinical-gated telemetry share flow or access to local app-container telemetry.
- Residual prior-participant data on a reused study phone (mitigated by the New Participant Reset wipe, 2026-07-14; between-participant lifecycle control in section 5).
- Data exfiltration via a compromised or misconfigured cloud upload path.
- Dependency vulnerability introducing runtime compromise.
- Account takeover or unauthorized API access via weak/misconfigured auth flows.
- Privilege escalation from incorrect role/scope mapping.
- Regression that restores Files sharing/open-in-place keys, weakens local file protection, or leaves a legacy Documents export behind after upgrade.
- Security-relevant behavior drift between local forks of `OmnibLe` / `G7SensorKit` / `LoopKit` and their upstream source without explicit provenance tracking.
- Tampering with locally persisted runtime recovery state (pod identity, delivery baselines, pending issued-dose attributions, fallback reconciliation records in `UserDefaults`) to distort masked-fallback or issued-dose replay accounting on the next launch (added 2026-07-09; behavior introduced by the 2026-06/07 fallback/replay/assumed-delivered feature set, RA-006/RA-008/RA-017).
- Divergence or tampering between the reviewed `Algo2015 C++` source and the locally built `Algo2015 / Algo2015Safety` xcframeworks the app ships - bounded by the build scripts' identity report (safety binary byte-identical to the primary except three renamed entry symbols), differential-replay characterization discipline for any input change, and dual-instance isolation regression tests.
- Tampering with or replaying provisioned clinical-unlock verifier material or its local single-use counter to gain clinical-settings access (added 2026-07-09; bounded by HMAC verification, counter burn before unlock, expiry, and lockout - see `SRS-CLIN-*`).

8. Cybersecurity Evidence Artifacts

Available controlled evidence:

- trust-boundary and threat-scenario analysis in sections 3, 5, and 7 of this plan
- supplier/inherited-control matrix in section 4 and the current provenance, embedded-package delta, dependency inventory, and supplier-request records
- local file/permission, logging/secret, and export/file-sharing reviews
- Build 843 CycloneDX SBOM and dated manual license/advisory review
- exact-freeze `TV-SEC-001` execution and composition/manual-advisory companion

- Build 843 dependency verification, archive identity, and retained release artifacts
- IDE cybersecurity closure matrix distinguishing complete, open, and deferred controls

Controlled source locators include:

- Docs/Quality/Cybersecurity_SOUP_Provenance_Review.md
- Docs/Quality/Cybersecurity_Embedded_Package_Delta_Review.md
- Docs/Quality/Cybersecurity_Local_File_and_Permission_Review.md
- Docs/Quality/Cybersecurity_Dependency_Inventory.md
- Docs/Quality/Cybersecurity_SBOM_and_Advisory_Process.md
- Docs/Quality/Cybersecurity_Logging_and_Secret_Review.md
- Docs/Quality/Cybersecurity_Supplier_Artifact_Request_List.md
- Docs/Quality/Cybersecurity_Baseline_Acceptability_Recommendation.md
- Docs/Quality/Cybersecurity_TV_SEC_001_Freeze_Execution_Checklist.md
- exact-freeze composition/manual-advisory companion:
Docs/Quality/Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-sbom-advisory-manual/README.md
- archive-826 matching CryptoSwift 1.9 impact companion retained as historical evidence:
Docs/Quality/Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-cryptoswift-1.9-companion/README.md
- controlled Build 843 CryptoSwift 1.10 verification:
Docs/Quality/Cybersecurity_CryptoSwift_1_10_Candidate_Verification.md
- Build 843 CycloneDX SBOM and dated review:
Docs/Quality/Cybersecurity_SBOM_Build_843.cdx.json and
Docs/Quality/IDE_Software_Packet/IDE_Build_843_SBOM_and_Advisory_Review.md
- IDE cybersecurity closure matrix:
Docs/Quality/IDE_Software_Packet/IDE_Cybersecurity_Closure_Matrix.md

Evidence still needed only if its associated scope or claim is retained:

- approved supplier/FDA support for relied-upon DASH and G7 security properties
- an approved automated dependency scanner output if the controlled manual SBOM/advisory method is not accepted
- penetration/resilience, identity-provider, and authentication negative-test evidence if the deferred cloud/authentication scope is brought into the IDE software claim
- incident-response drill evidence, update/patch governance, end-of-support, and vulnerability-monitoring records identified in section 10

9. Current Control and Evidence Matrix

Security Area	Related IDs	Current Submission Scope	Current Control / Design Statement	Verification / Evidence Position
Inherited DASH and G7 link security	RA-003, RA-009	Supporting inherited control context	Current code shows encrypted DASH transport and G7 authentication surfaces; BionicLoop relies on supplier/device security rather than re-implementing device firmware controls.	Supplier/FDA artifact linkage remains needed before submission closure can rely on the inherited controls.
Local telemetry export and file handling	SRS-SEC-001.002, SRS-LOG-015, TV-SEC-001, TV-LOG-015, RA-009, RA-024	In scope	The participant build omits Files sharing/open-in-place keys; retained telemetry, complete active-session step evidence, source Algo2015 artifacts, and recovery staging receive app-container protection/backup exclusion. A clinical-gated ZIP carries a bounded step CSV plus only matching/current epoch-session algorithm files.	Sponsor disposition is closed Low for RA-009. Exact-freeze TV-SEC-001 passed 68/68 scoped local-control tests. Any retained physical ZIP claim remains subject to the defined evidence-promotion process.
Secure cloud upload and study-data transport	SRS-SEC-001, TV-SEC-002, RA-009	Supportive function; broader formal verification deferred	The implementation supports secure cloud telemetry and a protected, clinical-gated local recovery export. Scout is supportive for monitoring, troubleshooting, corroborating software evidence, and sequence/gap review; local dosing does not depend on Scout availability.	Official study sources are Dexcom Clarity for CGM, the BionicLoop exported CSV for insulin delivery, and staff-entered electronic case-report forms for adverse events and other safety information. Commercial cloud, Part 11, and broader provider/authentication closure remain deferred.
Protected API access,	SRS-SEC-	Deferred	Existing implementation remains documented for software	No closure is claimed in this package; re-open only if the

Security Area	Related IDs	Current Submission Scope	Current Control / Design Statement	Verification / Evidence Position
onboarding providers, role authorization, password recovery, session restore, auth-failure continuity	003..009, TV-SEC-003..008, RA-009	from the current software package	understanding.	relied-upon study scope is widened.
SOUP / local-package provenance	RA-009 supporting control	In scope as documentation/processes requirement	Controlled <code>OmniBLE</code> , <code>G7SensorKit</code> , and <code>LoopKit</code> sources have recorded provenance, local-delta, dependency-inventory, and SBOM/advisory notes. The <code>Algo2015</code> binaries are built from separately controlled algorithm source; provenance is anchored by algorithm-source custody, controlled build scripts, and the safety-build identity report. Build 843 requires <code>CryptoSwift 1.10.0</code> exactly and fail-closed checks its source URL, version, and revision.	Exact-freeze and Build 843 companion records preserve the resolved dependencies, algorithm slice hashes, safety identity, advisory review, compatibility tests, archive identity, and physical transport observations. Build 843 was designated as the IDE submission build on 2026-09-03, closing <code>CYBER-DEV-001</code> ; evidence and cybersecurity acceptance were recorded under D08 and D09.

10. Current Missing Artifacts / Open Cyber Gaps

- Approved supplier or FDA-cited cybersecurity/interoperability artifacts for the exact Dexcom G7 and Omnipod DASH security properties being relied upon
- Any available upstream tag/release mapping for the identified `OmniBLE`, `G7SensorKit`, and `LoopKit` import/sync baselines
- Sponsor acceptance of the controlled manual SBOM/advisory method or execution of an approved scanner if the manual method is not accepted
- Continued retention of the Build 843 archive, installation, transport, and recovery-ZIP evidence accepted under D08 and D09
- Sponsor-approved software update and patch-delivery procedure for the investigational fleet, including authorization, installation, and re-verification triggers

- End-of-support statement for the investigational baseline (what happens to devices/data when the study build is superseded) (added 2026-07-09)
- Vulnerability-response cadence for the study window (monitoring source, triage SLA, and criteria for notifying the sponsor/IRB) beyond the incident-handling requirements in section 12 (added 2026-07-09)

11. Deferred Authentication Scope Note

For the current software package:

- SRS-SEC-003...009 are not claimed as closed software scope
- TV-SEC-003...008 are not claimed as submission evidence in this package
- SDD-AUTH-001, SDD-AUTH-002, SDD-POL-015, and SDD-POL-016 remain documented because the implementation exists, but they should be treated as documented implementation context rather than accepted closure claims for this package

BionicScout is approved as a supportive monitoring and corroborating evidence tool, not an authoritative dosing or study-outcome source. The approved scope, authoritative data sources, and deferred commercial cloud controls are recorded in IDE_Cloud_Telemetry_Submission_Scope_Decision.md.

12. Incident Response Requirements

The controlled post-approval software change procedure defines the required engineering assessment, risk and traceability updates, regression evidence, deployment authorization, study communication assessment, and sponsor-owned regulatory classification for a software incident or corrective change. The IDE Cybersecurity Closure Matrix retains the still-open sponsor decisions for named contacts, severity criteria, response timing, containment, recovery, vulnerability-monitoring cadence, and reporting responsibilities. Until those decisions are approved, this plan does not claim a completed incident-response SLA.

13. Regulatory Alignment

See Docs/Quality/RegulatoryReferences.md for FDA cybersecurity guidance links and applicability notes.

IDE Formal Evidence Index - 2026-08-21

Field	Value
Baseline	ide-software-freeze-2026-08-21 / 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Execution status	Complete
Organizational status	Evidence and traceability accepted under D08; cybersecurity scope and controls accepted under D09
Updated	2026-09-08

Formal Lane Index

Evidence bundle	Result	Open item
Evidence/Formal/STR-ALG-001/2026-08-21-ide-freeze-alg-baseline-r2/	Behavioral verification passed; Algo2015 line coverage 96.12%, branch coverage 87.77%; bridge line coverage 90.27%, branch coverage 65.85%; total-insulin host regression passed	Original execution retained; ALG-DEV-SA-006 closed by the companion below
Evidence/Formal/STR-ALG-001/2026-08-25-sa-006-linkage-companion/	Exact-freeze static-analysis/linkage companion passed 6/6; literal frozen SHA matched one review record and two SRS IDs	ALG-DEV-SA-006 closed
Evidence/Formal/STR-AUTO-001/2026-08-21-ide-freeze-auto-baseline-r2/	App target 997 total / 996 passed / 1 intentional IFU reference-table generation helper skip gated by an export directory / 0 failed; UI target 41 passed / 3 reproducible harness deviations	Original execution retained; AUTO-DEV-001 and AUTO-DEV-002 closed by the companion below
Evidence/Formal/STR-AUTO-001/2026-08-25-ui-harness-companion/	Corrected test-harness-only serial UI run passed 44/44	AUTO-DEV-001 and AUTO-DEV-002 closed
Evidence/Formal/STR-SIM-001/2026-08-21-101509-14034-ide-freeze-sim-baseline/	Core, pod, alert, and real-engine simulation lanes passed	Accepted under D08
Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-tv-sec-001/	68/68 scoped local controls passed	Accepted under D08 and D09

Evidence bundle	Result	Open item
Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-sbom-advisory-manual/	Composition, binary identity, source credential-pattern review, and controlled manual advisory evidence generated	Accepted within the cybersecurity scope under D09
Evidence/Formal/STR-SEC-001/2026-08-21-ide-freeze-cryptoswift-1.9-companion/	Archive-matching companion passed 68/68 scoped security, 8/8 OmniBLE crypto/session, 9/9 app unlock, and 19/19 core unlock tests	Historical archive-826 impact evidence; CYBER-DEV-001 closed 2026-09-03 by Build 843 designation

The archive-826 dependency mismatch is retained here as immutable execution history. CYBER-DEV-001 closed on 2026-09-03 when Build 843, using the controlled CryptoSwift 1.10.0 resolution, was designated as the IDE submission build. Build 843 dependency, archive, installation, and supporting real-Pod records are in Appendices A08A and A10A-C. Evidence and cybersecurity acceptance are recorded under D08 and D09 without rewriting the archive-826 history.

RTM Coverage

The canonical current-state matrix in RTM version 1.86 maps exact-freeze evidence to all 24 IDE-scope risk-analysis rows (RA-001 through RA-024). Evidence, traceability, residual risks, and cybersecurity scope are accepted under D07 through D09. The controlled multi-G7 exercise remains a predeployment condition under D06. The three previously open clinical policy questions are confirmed in Appendix A13A.

Evidence Classification

- Formal evidence: a controlled bundle under Evidence/Formal/ explicitly executed against or linked to the frozen baseline.
- Working evidence: development output not promoted into a controlled formal bundle. It may support investigation but is not formal evidence.
- Physical/operator evidence: formal only after the retained protocol, installed build, subject/device, operator, UTC window, result, signature, and sponsor-designated acceptance are complete.

The Freeze Execution Report is the authority if a summary conflicts with an individual bundle.

Cybersecurity Dependency Inventory

Status: Controlled dependency inventory accepted within the Build 843 cybersecurity scope

Owner: Software Developer Last updated: 2026-09-08

1. Purpose

Record the current software-composition baseline visible from the checked-in BionicLoop project and local package manifests.

This inventory is a controlled dependency snapshot. The exact-freeze composition and manual-advisory companion now exists at 2026-08-21-ide-freeze-sbom-advisory-manual.

Build 843 requires CryptoSwift 1.10.0 exactly, tracks the app-workspace resolution, and uses a fail-closed checker to enforce source, version, and revision. Historical exact-freeze dependency details remain in the Freeze Execution Report and its referenced evidence; this inventory presents the current controlled build composition. The Build 843 machine-readable source is `Cybersecurity_SBOM_Build_843.cdx.json`, with the dated review boundary in Technical Appendix A08B.

2. Scope of This Snapshot

Included here:

- BionicLoop app project dependency surfaces visible in the checked-in `BionicLoop.xcodeproj`
- local embedded packages and subprojects shipped with the app baseline
- remote SwiftPM packages resolved for Build 843, plus their source requirements in local manifests and projects
- binary algorithm artifact referenced by the app/core baseline

Operational follow-through:

- retain required third-party acknowledgments with the distributed materials
- continue vulnerability monitoring during the study; the controlled manual advisory method and its limitations are accepted under D09

3. Current Dependency Snapshot

3.1 First-party / repo-local components in the app baseline

Component	Integration Form	Current Baseline Evidence	Notes
BionicLoop app target	First-party app target	BionicLoop.xcodeproj/project.pbxproj	Primary shipped controller application.
BionicLoopCore	Local Swift package	BionicLoop.xcodeproj/project.pbxproj, BionicLoopCore/Package.swift	Local core domain/runtime package referenced through XCLocalSwiftPackageReference.
LoopKit	Local Swift package	BionicLoop.xcodeproj/project.pbxproj, LoopKit/Package.swift	Local package referenced through XCLocalSwiftPackageReference.
G7SensorKit / G7SensorKitUI	Local embedded subproject/framework	BionicLoop.xcodeproj/project.pbxproj references G7SensorKit.xcodeproj; upstream https://github.com/LoopKit/G7SensorKit ; MIT license verified from upstream LICENSE on 2026-09-03; traced sync base 624bb360c277c43daa5232df986de7467bc0b72b	Integrated as a checked-in local Xcode subproject rather than a remote package. The monorepo does not contain the upstream license file; provenance and license identity are recorded here without asserting current-tree identity to upstream HEAD.
OmniBLE	Local embedded subproject/framework	BionicLoop.xcodeproj/project.pbxproj references OmniBLE.xcodeproj	Integrated as a checked-in local Xcode subproject rather than a remote package.

Component	Integration Form	Current Baseline Evidence	Notes
Algo2015.xcframework	Local binary artifact	BionicLoop.xcodeproj/project.pbproj, BionicLoopCore/Package.swift	Binary algorithm artifact referenced from Algo2015/build/Algo2015.xcframework.
Algo2015Bridge	Local bridge target	BionicLoopCore/Package.swift	Repo-local bridge between host Swift code and Algo2015 binary artifact.
Algo2015Safety.xcframework	Local binary artifact	BionicLoopCore binary-target declaration and the controlled safety build/identity process	Symbol-isolated safety-controller artifact; the controlled identity report records its relationship to the primary algorithm compilation.
Algo2015SafetyBridge	Local bridge target	BionicLoopCore/Package.swift	Repo-local bridge between host Swift code and the isolated safety binary.

3.2 Remote SwiftPM dependencies observed at freeze

Packag e	Source	Resolved Version / Revision	Evidence	Current Use Surface
CryptoSwift	https://github.com/krzysztof-anowski/CryptoSwift	1.10.0 / f2a627b84c1ff96f21ac2fcb623ab36142dd5512	Build 843 tracked resolution and dependency verification	Used by OmniBLE cryptographic/session paths; exact source, version, and revision are checked before build.
SlideButton	https://github.com/no-comment/SlideButton	branch main at 5eacebba4d7deeb693592bc9a62ab2d2181e133b (1.3.0)	Formal-run console and captured resolution	Used for slider-style controls. Branch requirement remains unpinned by the tag.
SwiftCharts	https://github.com/ivanschuetz/SwiftCharts	branch master at c354c1945bb35a1f01b665b22474f6db28cba4a2	Formal-run console, captured resolution, LoopKit/Package.swift	Used through LoopKitUI; upstream states the project is no longer maintained. Branch requirement remains unpinned by the tag.

3.3 External regulatory context and non-reliance

The FDA 510(k) summary for Tidepool Loop, K203689, records that Tidepool evaluated known use problems from DIY Loop, adapted its design, and performed detailed verification and validation under the interoperable automated glycemic-controller requirements. This provides general regulatory context for bringing software with DIY Loop origins under formal design controls.

That clearance is not verification of BionicLoop's exact `OmniBLE` or `G7SensorKit` source, local modifications, or Build 843 device interfaces. No component-equivalence or inherited-verification claim is made. BionicLoop relies on the controlled Build 843 source and its own requirements, risk management, configuration controls, verification, and supporting real-device observations. See the [FDA Tidepool Loop 510\(k\) summary](#).

4. Current Observations

- The app project itself currently uses local `BionicLoopCore` and local `LoopKit` package references, plus local `G7SensorKit` and `OmniBLE` subprojects.
- Build 843 tracks the app-workspace `Package.resolved` file and pins `CryptoSwift` exactly.
- `LoopKit` also records `SwiftCharts` in its own package manifest and lockfile.
- `BionicLoopCore` depends on two local binary artifacts (`Algo2015.xcframework` and `Algo2015Safety.xcframework`) in addition to local bridge/Swift targets.
- The standalone `LoopKit.xcodeproj` development workspace lockfile records `SwiftCharts` revision `3d011f67eccb1ffa622fbfccb1348eed80309ae8`, while the observed BionicLoop app workspace and `LoopKit/Package.resolved` record `c354c1945bb35a1f01b665b22474f6db28cba4a2`. The BionicLoop build resolves local `LoopKit` through the app workspace, so the standalone development workspace lockfile is not a shipped-baseline input and is excluded from the app SBOM. It must not be substituted for the shipping app resolution during freeze capture.

4.1 Exact-freeze formal-run binary checksum snapshot

These SHA-256 values were regenerated during the exact-freeze formal run. The controlled text manifest is `binary-checksums.txt`.

Artifact Slice	SHA-256
<code>Algo2015.xcframework/ios-arm64/libAlgo2015.a</code>	<code>b8b6cb918882b9ae40f0dacb1e4cc235ac374f2bf3ecfbe4d6882a0903211363</code>
<code>Algo2015.xcframework/ios-arm64-simulator/libAlgo2015.a</code>	<code>ef12558672946a70911279bb947d29c81f5bdb6b5822b807b0291ab1dd5de119</code>
<code>Algo2015.xcframework/macos-arm64_x86_64/libAlgo2015.a</code>	<code>97512f06d0309407884b71cf88782fb67cc790421f47771c10c708bbef3dfa2e</code>

Artifact Slice	SHA-256
Algo2015Safety.xcframework/ios-arm64/libAlgo2015Safety.a	5a565f8e71ff0f9b7ffefab3ec895ec91f4e36f34a365648788e87cff5e29cac
Algo2015Safety.xcframework/ios-arm64-simulator/libAlgo2015Safety.a	6fa9470cdb05eecb6cbaadd71c5a5dbb354cb2c4a5f8c0dd953407e954337987
Algo2015Safety.xcframework/macos-arm64_x86_64/libAlgo2015Safety.a	050b7bb981241fcdde6450b996c1c172941fbee33bc8325d5d0e32d85ebfe5c6

The exact-freeze safety identity report records byte-identical primary/safety objects before the three intentional exported-symbol renames.

4.2 Exact-freeze controlled manual advisory review

- The dated repository and GitHub global Swift advisory queries in advisory-review.md returned no published advisory requiring an immediate source change.
- The same review records SwiftCharts' explicit unmaintained status as a maintenance residual rather than treating "no advisory" as active support.
- `gitleaks`, `trufflehog`, `syft`, `grype`, and `osv-scanner` were unavailable. The controlled manual method and its limitations are accepted under D09; an approved scanner may be added later as an operational enhancement.
- The Build 843 dependency set passed the dependency checker and its 6/6 unit tests, 8/8 OmniBLE cryptographic/session vectors, 471/471 core tests, the canonical 997-test app run with zero failures and one intentional IFU reference-table generation helper skip gated by an export directory, and 68/68 scoped security controls. Build 843 archive identity is recorded; evidence and cybersecurity acceptance are recorded under D08 and D09.

5. Current Limitations

This inventory retains these operational items:

- required distributed acknowledgment placement
- recurring vulnerability-monitoring ownership and cadence
- any future decision to supplement the accepted manual method with an organization-approved scanner

6. Current Recommendation

For the current software package:

- use this document as the current dependency-inventory floor
- preserve exact-freeze dependency history in its execution record and retain the tracked app-workspace resolution plus fail-closed checker for Build 843 and future baselines

- use Cybersecurity_SBOM_and_Advisory_Process.md as the current process/ownership note
- use Cybersecurity_SBOM_Build_843.cdx.json as the machine-readable Build 843 composition record and Technical Appendix A08B as its dated license/advisory review
- use the exact-freeze companion as the current engineering composition, checksum, identity, secret-pattern, and manual-advisory record
- retain the recorded Build 843 archive, installation, and focused real-Pod transport evidence
- obtain sponsor evidence disposition before final promotion and retain exact app-workspace resolution control in every future baseline

Build 843 CryptoSwift 1.10 Controlled-Source Verification

Status: Controlled Build 843 engineering evidence accepted under D08 and D09

Controlled Source Identity

Field	Value
Frozen baseline retained as history	ide-software-freeze-2026-08-21 / 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Dependency-control commit	a4cff45a40f740fa3624f2015bfb3e9bfb781211
Verified build-input commit	77c26813a6a9bd269b623ba22df43bf7af05fac8
Signed archive source	walkabout-candidate-2026-08-21-build-843 / 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Signed archive identity	BionicLoop 1.0 (843) / CC-DEV / SHA-256 80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
CryptoSwift source	https://github.com/krzyzanowskim/CryptoSwift.git
Exact version / revision	1.10.0 / f2a627b84c1ff96f21ac2fcb623ab36142dd5512
App-workspace resolution-manifest SHA-256	019b5768cfb4fc5c4c3c27106b7df9d60e0d7988eed34a3e75f4e523c625ed
Other remote resolutions	SlideButton 5eacebba4d7deeb693592bc9a62ab2d2181e133b; SwiftCharts c354c1945bb35a1f01b665b22474f6db28cba4a2 (unchanged)

The OmniBLE project now requires CryptoSwift 1.10.0 exactly. The app-workspace Package.resolved file is tracked and a fail-closed checker verifies the source URL, exact project requirement, resolved version and revision, Git visibility, and tracked-file status.

This controlled source does not modify the immutable freeze tag, archive 826, or their formal evidence. It was used to produce signed archive 843, whose release and installation records are controlled separately.

TDD and Compatibility Finding

The dependency-control regression first failed against the prior ranged requirement and ignored 1.9.0 workspace resolution, then passed after the exact requirement and tracked lockfile were added.

The first CryptoSwift 1.10 OmniBLE test build exposed a test-harness-only Swift 6 compatibility issue: `Data.bytes` is a `RawSpan`. Commit 77c26813 applies the already-characterized `Data.byteArray` bridge in one packet test. No product source, cryptographic operation, expected vector, or assertion meaning changed.

Verification Results

Verification	Result
Dependency-control checker	Pass
Dependency-control unit tests	6/6 passed
Clean Xcode package resolution	Pass; CryptoSwift 1.10.0 at the controlled revision
OmniBLE encryption/session vectors	8/8 passed
BionicLoopCore	471/471 passed
Canonical serial BionicLoopTests	997 executed; 996 passed; 1 intentional IFU reference-table generation helper skip gated by an export directory; 0 failed
Scoped TV-SEC-001 controls	68/68 passed
Simulation merge gate	Pass; no high-risk runtime path changed
SwiftFormat	0/358 files require formatting
SwiftLint	Existing repository baseline: 168 findings / 27 serious; no finding in the touched test file
Xcode Analyze	Pass; pre-existing warnings only
Clean simulator app build	Pass; app 1.0 / build 839; not a designated archive
Clean-checkout signed archive	Pass; app 1.0 / build 843; CryptoSwift 1.10.0; strict code-sign verification passed

Verification methods:

- fail-closed dependency-control checker and its unit suite
- BionicLoopCore package tests
- canonical serial BionicLoop app tests
- risk-based simulation merge gate
- SwiftFormat and SwiftLint quality checks
- Xcode static analysis

Evidence Classification and Disposition

These controlled Build 843 engineering results support the 2026-09-03 build designation and closure of CYBER-DEV-001. Their submission-package acceptance is recorded under D08 and D09.

Archive 843 is recorded in

IDE_Software_Packet/BionicLoop_1_0_843_Tester_Release_Record.md. App Store Connect accepted the archive and its symbols. The archive, dSYMs, export options, and raw distribution logs are retained under the checksum-controlled locator recorded in that release record. Build 843 installation and focused real-Pod use are recorded as supporting evidence. Build 843 is the approved IDE submission build, and CC-DEV is its confirmed participant Scout service environment under D12. Scout remains supportive and does not affect local dosing. The controlled Build 843 Working Hardware Evidence Review is Appendix A10B.

Build 843 SBOM and Advisory Review

Field	Value
Build	BionicLoop 1.0 (843), source 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Archive SHA-256	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
Machine-readable SBOM	Cybersecurity_SBOM_Build_843.cdx.json, CycloneDX 1.5
Review date	2026-09-03
Method	Controlled manifest comparison, local license inspection, and dated GitHub repository/advisory review
Status	Engineering review and listed limitations accepted within D09; operational follow-up retained

Composition Result

The machine-readable SBOM records the Build 843 application, repository-local packages, both Algo2015 binaries, and all three Swift Package Manager dependencies. The external pins exactly match the controlled `Package.resolved` file:

Component	Build 843 identity	License review
CryptoSwift	1.10.0 / f2a627b84c1ff96f21ac2fcb623ab36142dd5512	Project attribution license; required acknowledgment must be retained
SlideButton	5eacebba4d7deeb693592bc9a62ab2d2181e133b	MIT
SwiftCharts	c354c1945bb35a1f01b665b22474f6db28cba4a2	Apache-2.0
LoopKit	repository-local Build 843 source	MIT
OmniBLE	repository-local Build 843 source	MIT
G7SensorKit	repository-local Build 843 source; traced upstream sync base 624bb360c277c43daa5232df986de7467bc0b72b from https://github.com/LoopKit/G7SensorKit	MIT license verified from the upstream LICENSE file on 2026-09-03; the monorepo does not contain a local license copy

Component	Build 843 identity	License review
Algo2015 / Algo2015 Safety	sponsor source b855f871c80e95d6f9226f54b0e60168ff603e36	Separately controlled sponsor source; external open-source license not asserted

Dated Advisory Review

On 2026-08-27, GitHub's Swift Advisory Database was queried for CryptoSwift 1.10.0, SlideButton, and SwiftCharts at their controlled identities. No matching advisories were returned. CryptoSwift 1.10.0 was also confirmed as the upstream release associated with commit `f2a627b`.

This is a dated public-database review, not output from an organization-approved software-composition scanner. It does not establish that no vulnerability exists. Repository-local forks, proprietary device behavior, iOS, and the separately controlled algorithm require their own monitoring and supplier evidence boundary.

Operational Follow-Up

- retain the accepted controlled manual method and consider an approved scanner as a future operational enhancement
- retain the recorded G7SensorKit upstream/license attribution and decide the distributed acknowledgment placement
- retain the CryptoSwift acknowledgment in distributed documentation
- establish the owner and cadence for vulnerability monitoring during the investigation
- assess whether newly identified vulnerabilities require a software change, labeling/consent update, IDE report, or emergency action

The IDE Cybersecurity Closure Matrix tracks these actions without presenting the Build 843 dependency verification as broader cybersecurity closure.

IDE Cybersecurity Closure Matrix

Field	Value
Baseline	Frozen source 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e; designated Build 843 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Purpose	Concise disposition of study-used cybersecurity controls and operational follow-through
Status	Cybersecurity scope and controls accepted under D09; operational items remain where shown
Date	2026-09-08

Control area	Current state	Remaining action/disposition
Protected local clinical records, migration, backup exclusion, and gated export	TV-SEC-001 passed 68/68 at the frozen baseline	Complete; accepted under D08 and D09
Build 843 dependency resolution	Exact CryptoSwift 1.10.0 and controlled Package.resolved; dependency and crypto/session tests passed	Complete; Build 843 approved for IDE submission 2026-09-03
Machine-readable SBOM	CycloneDX 1.5 SBOM generated and reconciled to the Build 843 manifest	Complete; accepted within D09
Vulnerability review	Dated public GitHub advisory review found no matching external-package advisories	Accepted manual method for submission; define operational monitoring cadence
License review	CryptoSwift attribution, SlideButton MIT, SwiftCharts Apache-2.0, LoopKit/OmniBLE MIT, and G7SensorKit MIT with traced upstream sync base identified	Accepted within D09; retain required distributed acknowledgments
Threat modeling	Current plan lists study-relevant scenarios and trust boundaries	Open; complete and approve a structured worksheet if required by sponsor process
G7 and DASH inherited controls	Device/interface boundaries and local implementation are documented	Open; retain approved supplier/FDA-cleared-product references for relied-upon controls
Update and patch process	Post-approval software change procedure exists	Open; approve owner, severity criteria, deployment path, and study communication process

Control area	Current state	Remaining action/disposition
Vulnerability monitoring	No approved recurring cadence is recorded	Open; assign owner, sources, frequency, and escalation criteria
Incident response	General severity and response expectations are documented	Open; approve contacts, timing, containment, reporting, and recovery procedure
Cyber labeling and informed-consent assessment	Investigational and device-use limitations are present	Open; Sponsor/Regulatory determines any cyber-specific labeling or consent content
Scout telemetry	Supportive only; local dosing does not depend on Scout; local recovery ZIP is primary software reconstruction evidence	Approved scope; retain corroborating sequence records and investigate material gaps; broader cloud/auth/Part 11 deferred
Clinical Settings unlock	Subject-scoped, single-use offline verification and secure local state are implemented and tested	Complete; accepted under D08 and D09
Broader provider/authentication and commercial cloud	Not relied upon for local dosing or official study outcomes under the approved Scout scope	Deferred; reopen only if relied-upon study scope expands

Final cybersecurity acceptance is recorded under D09. This matrix preserves the operational follow-through without reopening closed Build 843 dependency work or expanding the accepted supportive-Scout boundary.

IDE Software Version and Configuration Identification

Status: Engineering configuration record; Build 843 approved for IDE submission and evidence accepted Updated: 2026-09-08

This record separates the frozen source, exact-freeze verification inputs, signed Build 843 tester archive, eventual deployable study build, and documentation package. These identities must not be treated as interchangeable.

1. Frozen Source Baseline

Field	Controlled identity
Source baseline tag	ide-software-freeze-2026-08-21
Source baseline commit	91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Last shipping-product commit in that baseline	ae00e754
Marketing version	1.0
Status	Immutable engineering software-source freeze; not by itself a deployable-build designation or sponsor approval

Documentation-only commits after the source freeze do not alter this product baseline. Any product source, build configuration, algorithm input, or dependency-resolution change requires a new configuration identity and an impact-based rerun decision.

2. Exact-Freeze Verification Inputs

Item	Identity
Toolchain	Xcode 26.6 (17F113), macOS 26.6.2 (25G83)
CryptoSwift	1.10.0 at f2a627b84c1ff96f21ac2fcb623ab36142dd5512
SlideButton	5eacebba4d7deeb693592bc9a62ab2d2181e133b
SwiftCharts	c354c1945bb35a1f01b665b22474f6db28cba4a2
Resolution-manifest SHA-256	019b5768cfb4fc5c4c3c27106b7df9d60e0d7988eed34a3e75f4e523c625ede d

Item	Identity
Evidence authority	IDE_Freeze_Execution_Report_2026-08-21.md and its referenced formal evidence bundles

Algorithm Inputs

Item	Identity
Algo2015 source repository commit	b855f871c80e95d6f9226f54b0e60168ff603e36
Algorithm_2015_10_13.cpp	SHA-256 f2c61ddc46c45f40bd53712610e00060105d6fe4d912d25f61b34118dc10081e
AlgorithmInterface.h	SHA-256 392970bd82f8cc5d978131e1118bb0288e95c57875e469092c8f6c9bcc80368f
ringbuffer.h	SHA-256 869bacce22e01cb979bfeae4bf1f31c239b2f93689cc639d3cf305c46b6b8ab3
Primary/safety relationship	Two symbol-isolated compilations of the same unmodified source; identity evidence is retained in STR-ALG-001 and the composition companion

Build Configuration Inputs

Item	SHA-256
Base.xcconfig	ed7d66f263f626e94e703538945ad63852b1792fbb739ee936c5e59ac38f9e2c
Debug.xcconfig	df4c78aa7ea8c5e7f0d2a4981e17dcbf1bc856ddce5671f918767f384a12a792
Release.xcconfig	8c83ecc3acd2c23a2fcde93b2aa4e83329e9811d96a31b58bb47149cc55fb942
CC-DEV.xcconfig	9c50fa680863be12da1bd604695c6bfe0a2b6b79c5be44d519b077f777d34c80

3. Signed Build 843 Archive

Field	Controlled identity
Source tag	walkabout-candidate-2026-08-21-build-843
Source commit	1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Archive identity	BionicLoop 1.0 / build 843 / org.bionicleop.app / CC-DEV
Archive package SHA-256	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
CryptoSwift	Exact 1.10.0 at f2a627b84c1ff96f21ac2fcb623ab36142dd5512

Field	Controlled identity
Compiled Scout service environment	CC-DEV; supportive service only, with no effect on local dosing
Retained artifact locator	ReleaseArtifacts/Build-843/; Git-excluded archive, dSYMs, export options, and raw upload logs, checksummed in the tester release record
Status	App Store Connect upload, installation, real-Pod supporting observation, 2026-09-03 IDE-submission approval, and D08 evidence acceptance complete

Build 843 includes tracked build-configuration inputs and an exactly resolved dependency manifest. It passed strict code-sign verification. Full details are in `BionicLoop_1_0_843_Tester_Release_Record.md`.

Archive 843 is the controlled build approved for IDE submission. CC-DEV is the Scout service environment compiled into Build 843 through `CC-DEV.xcconfig`. Scout is supportive and does not affect local dosing. A different service environment requires a new build identity and corresponding delta assessment. CC-DEV is confirmed for participant use under D12.

4. IDE Submission Build

Status: Build 843 approved for IDE submission on 2026-09-03. Source identity, archive number, and installed build number remain separate controlled identities.

The controlled record for the designated build includes:

- archive/build number and immutable archive checksum
- source tag and full source commit
- resolved dependency manifest and build-configuration checksums
- signing identity, entitlements, provisioning profile, and distribution record
- device/subject assignment and install confirmation
- relationship to the exact-freeze evidence
- any required regression or companion execution
- Sponsor/Regulatory and designated software-evidence approval

Apple accepted archive 843 with its symbols, and installed Build 843 identity plus real-Pod supporting evidence are recorded in Appendix A10B. Evidence, residual-risk, and final package acceptance are recorded under D07, D08, and D11.

5. Documentation and Package Identity

Controlled document	Revision
Risk Analysis	1.53

Controlled document	Revision
Software Requirements Specification	1.82
Software Design Description	1.84
Software Verification and Validation Plan	1.91
Requirements Traceability Matrix	1.86
Cybersecurity Plan	1.03
IFU-BL-001	Revision 1.17; approved for IDE submission 2026-09-04

Package identity is recorded through a reproducible two-stage process so no document is required to contain its own hash:

1. **Stage 1 - source correction commit.** Commit the controlled Markdown, manifests, build script, and styles. This immutable commit is the package source commit.
2. **Stage 2 - rendered-artifact manifest.** Build both PDFs from the exact Stage-1 commit, then create `IDE_Software_Package_Release_Manifest_2026-08-21.md` recording the Stage-1 source commit, controlled document revisions, PDF filenames, SHA-256 hashes, generation timestamp, draft/release status, and approval state. Commit the manifest and generated PDFs separately.

The Stage-2 manifest is the authoritative package identity. A draft package must not be represented as approved or submission-ready merely because its hashes are recorded.

BionicLoop 1.0 (843) Tester Release Record

Updated: 2026-09-08

Status: App Store Connect upload, installation, and IDE-submission approval complete; focused real-Pod use observed as supporting evidence; evidence accepted under D08

Source Identity

Field	Recorded value
Source tag	walkabout-candidate-2026-08-21-build-843
Source commit	1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Marketing version	1.0
Build number	843
Bundle identifier	org.bionicloop.app
Release environment	CC-DEV

CC-DEV is the Scout service environment compiled into Build 843 through CC-DEV.xcconfig. Scout is supportive and does not affect local dosing. Build 843 was separately approved for IDE submission on 2026-09-03. A different service environment requires a new build identity and corresponding delta assessment. The CC-DEV participant service environment is confirmed under D12.

The archive was built from a detached clean checkout of the source commit. The ignored external Algo2015 repository was checked out at controlled commit

b855f871c80e95d6f9226f54b0e60168ff603e36; the primary and safety XCFrameworks were rebuilt before archiving, and the safety identity check reported identical primary compilation objects for iOS and iOS Simulator.

Controlled Inputs

Input	SHA-256
App-workspace Package.resolved	019b5768cfb4fc5c4c3c27106b7df9d60e0d7988eed34a3e75f4e523c625eded
Base.xcconfig	ed7d66f263f626e94e703538945ad63852b1792fbb739ee936c5e59ac38f9e2c

Input	SHA-256
Debug.xcconfig	df4c78aa7ea8c5e7f0d2a4981e17dcbf1bc856ddce5671f918767f384a12a792
Release.xcconfig	8c83ecc3acd2c23a2fcde93b2aa4e83329e9811d96a31b58bb47149cc55fb942
CC-DEV.xcconfig	9c50fa680863be12da1bd604695c6bfe0a2b6b79c5be44d519b077f777d34c80
CryptoSwift	1.10.0 / revision f2a627b84c1ff96f21ac2fcb623ab36142dd5512

Release preparation found that `Base.xcconfig` and `Debug.xcconfig` had been present during verification but ignored by the repository's broad `build/` rule. Commit `1455cda7` tracks the unchanged, non-secret files. Their hashes match the exact-freeze configuration record. This correction changes source reproducibility, not application behavior or effective build settings.

Archive Identity

Field	Recorded value
Archive filename	BionicLoop-1.0-843.xcarchive
Immutable archive package	BionicLoop-1.0-843.xcarchive.zip
Archive package SHA-256	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
Archive package size	50,773,805 bytes
Embedded app identity	org.bionicloop.app / 1.0 / 843 / CC-DEV
Archive signing identity	Apple Development: John Costik (V9G3KJL3HW)
Team identifier	P964PFM8CF
Provisioning profile	iOS Team Provisioning Profile: org.bionicloop.app
Provisioning UUID	11df9386-b6ac-42bc-8248-8f6f6e87542f
Provisioning expiration	2027-06-10
Code-directory hash	8c1d1bab8b4e2aaa5f359ef9067fad1b2f86b962
Archive result	ARCHIVE SUCCEEDED; deep strict code-sign verification passed

App Store Connect export applied managed distribution signing using `Apple Distribution: John Costik (P964PFM8CF)`. The upload options explicitly enabled symbol upload; the distribution pipeline generated and uploaded the archive symbol payloads with the application package.

App Store Connect Upload Record

Field	Recorded value
Upload result	Upload succeeded; Uploaded BionicLoop; EXPORT SUCCEEDED
Upload accepted	2026-08-21 16:00:44 EDT / 2026-08-21 20:00:44 UTC
App Store Connect state at original upload handback	Uploaded package was processing; subsequent Build 843 installation is confirmed below
Distribution method	App Store Connect upload
Distribution signing identity	Apple Distribution: John Costik (P964PFM8CF)
Distribution certificate SHA-1	061D78329ED47DADAB8D5C14C47E1AF0BA693F7A
Symbol upload	Enabled; distribution pipeline generated and uploaded symbol payloads

Apple acceptance of the upload is not equivalent to TestFlight processing, tester-group assignment, installation, or sponsor designation.

Retained Release Artifacts

Controlled storage locator: `ReleaseArtifacts/Build-843/` relative to the repository root. The build-specific directory is intentionally excluded from Git; `ReleaseArtifacts/README.md` defines the local retention rule. The large binary artifacts are not committed to the repository.

Retained artifact	Bytes	SHA-256
BionicLoop-1.0-843.xcarchive.zip	50,773,805	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
BionicLoop-1.0-843.dSYMs.zip	31,452,797	2e3f2040ef45d3f6397723cb1bb5789dc28f0b48dee5d89c8b6ec48ff4b54152
BionicLoop-1.0-843-ExportOptions.plist	646	ed42eb4a3d30780b5e0864a84dd0cfd3dc4788c517f58bce6a911b40866f9d2
BionicLoop-1.0-843.xcdistributionlogs.zip	137,809	174f428e90e7dfb1abd6654c449887598374df445e07bbaf11066188d6c737ee

The retained archive ZIP is byte-identical to the upload source and contains the original archive dSYM bundles. The separate dSYM ZIP is a retained access copy of that archive's dSYMs/ directory. The distribution-log ZIP contains the complete raw Xcode .xcdistributionlogs

bundle for the accepted upload. All three retained ZIPs passed archive-integrity inspection. Promotion or backup of this local vault into the sponsor's controlled evidence system remains a separate disposition; local retention does not make the artifacts formal evidence.

Distribution and Installation Status

Gate	Status
Source commit and tag published	Complete
Signed Release archive	Complete
App Store Connect upload	Complete; accepted 2026-08-21 20:00:44 UTC
TestFlight processing	Complete; Build 843 was installed and emitted Build 843 device evidence
Tester-group assignment record	Not separately retained in this repository
Sponsor-device install identity	Complete as working evidence: subject JRC-FORMAL-VV, Build 1.0 (843), CC-DEV, window 2026-08-24 18:30Z through 2026-08-25 15:11Z
Focused real-Pod encrypted session/command observation	Complete as working evidence; normal commands, feedback, outage recovery, suspend/resume, and continued session operation were observed

The focused Pod observation is recorded in the Build 843 Working Hardware Evidence Review. The app retained the active Pod/session, received fresh status, applied normal algorithm-issued commands, accepted pump feedback, and completed same-Pod recovery and suspend/resume operation. The retained working bundle includes an exact-session recovery ZIP and corroborating telemetry. Phone model/iOS, operator signature/date remain absent, so this observation is not represented as a separate formal operator-validation claim.

Evidence Classification

This is a controlled engineering release record. Build 843 uses an exactly tracked dependency resolution and is the approved IDE submission build by controlled approval dated 2026-09-03. Evidence and residual-risk acceptance are recorded under D08 and D07.

Build 843 Working Hardware Evidence Review

Field	Value
Classification	Controlled review of working/support evidence; not formal verification evidence
Subject	JRC-FORMAL-VV
App	BionicLoop 1.0 (843) / CC-DEV
Source	walkabout-candidate-2026-08-21-build-843 / 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Exercise window	2026-08-24 14:30 EDT through 2026-08-25 11:10:29 EDT
Review date	2026-08-27
Updated	2026-09-08
Sponsor status	Supporting evidence accepted under D08; no separate formal hardware-validation claim

Purpose And Baseline Boundary

This review correlates the completed Build 843 formative real-device evaluation with the device-local recovery export, operator observations, and BionicScout records. It supports an impact and evidence-reuse decision for the intended IDE study build.

Build 843 is not the same configuration identity as engineering freeze `ide-software-freeze-2026-08-21` at `91c0e98a9bc9429a0486bebddebffc7d8dbbe300e`. Exact-freeze formal evidence remains governed by the 2026-08-21 Freeze Execution Report. This review did not itself designate Build 843 and does not promote working evidence to formal evidence; the separate designation was recorded on 2026-09-03 in Appendices A10C and A18.

Evidence Hierarchy

For software reconstruction of the tested algorithm session, the preferred working-evidence order is:

1. the clinical-gated, device-local recovery ZIP containing the protected step CSV and byte-exact files for the selected Algo2015 session
2. controlled automated evidence bundles for requirements exercised by those suites
3. the operator record for visible UI, notification, and physical handling observations

4. BionicScout/Dynamo and app-log records as corroborating backup, sequence correlation, and gap investigation

Scout is not required for local algorithm execution or dosing. Official study sources are Dexcom Clarity for CGM outcomes, the BionicLoop exported CSV for insulin-delivery outcomes, and staff-entered electronic CRFs for adverse events and other safety information.

The available ZIP was exported at 09:38:30 EDT on 2026-08-25. It is therefore primary local evidence only through its 09:33:48 EDT final step. Temporary Target and suspension exercises after that cutoff currently rely on the operator record plus corroborating telemetry. A post-run ZIP is required if a single local export is expected to cover those later steps.

Retained Device-Local Export

The controlled source repository retains the supporting ZIP and extracted-file hashes under the Build 843 engineering evidence record.

Artifact	Size	SHA-256
JRC-FORMAL-VV-algo-files-20260825-093830.zip	297,010 bytes	446914e79d535f32c646e003a8ddc639b6dde661838cfe1eced74a44f8cec409
BionicLoop_StepTelemetry.csv	99,327 bytes	77f7d00a1dd65705faf49f0bdb920a676fa499b9e756a3f10838b9aebf8a1067
BP_1787597333.txt	69,425 bytes	3fdad25b0bda38eaf993c1aaf3210497ef330e32f791c1f717a617b02fce234e
BP_LOG_1787597333.txt	118,181 bytes	52b0c2fa9952848b9816692c051a642e8cbe131cb822ec27f9a5fdab27dc5ba4
BP_ReadMatrix.m	9,589 bytes	e84eec770f5d1fc2fcd4f39559b8076dcad46b3bc81c45f2e954799eb3bb4d5a

The ZIP opens, contains four nonempty files, and contains one algorithm epoch. Epoch 1787597333 is 2026-08-24 14:48:53 EDT and matches the session start. The CSV contains 222 data rows plus its header, from step 0 at 14:48:52 EDT through step 225 at 09:33:48 EDT. No unrelated algorithm session was included.

The CSV records manual BG consumption at step 7 (133 mg/dL) and step 164 (99 mg/dL). It also covers the ordinary-use, meal, relaunch, extended-CGM- outage, same-Pod fallback-recovery, and canceled-partial-meal portions performed before the export cutoff.

Deterministic Checker And Operator Correlation

The bounded checker run used signature set 0.13 and checker commit 564f912f over 2026-08-24 18:30:00Z through 2026-08-25 15:11:00Z. It reviewed 7,510 timeline events and 6,105 app-log lines, all identified as Build 843.

Exercise ID	Checker result	Correlated disposition
INT-01	Partial	App logs are present, but the expected <code>remote_logging_config_applied</code> marker is absent; tool/config evidence disposition required
INT-02	Pass	Readings present; no discrepancy reported
INT-03	Pass	Algorithm session continued through relaunches
INT-04	Pass	Meal delivery/progress cleared; commands and feedback present
INT-05	Missing by frozen signature	Manual-BG submission used <code>bg.modal.submit_button</code> ; signature 0.13 incorrectly searches for <code>home.bg_button</code> on the submit event. Operator and ZIP prove entry and step consumption. Preserve as an execution-tool deviation rather than changing the signature after execution.
INT-06	Pass	CGM-outage guidance, reminder, backup-basal bridge, BG recovery, and resumed stepping correlated
INT-07	Not executed	Conditional Pod-change exercise; no Pod change occurred in the bounded run
INT-08	Pass	Four launches, same algorithm session, no step 0 or stuck meal state
INT-09	Pass	Operator confirmed expected notification presentation; scheduled-notification evidence present
INT-10	Pass	Short Pod signal-loss alert progression and recovery were expected
INT-11	Pass	120 mg/dL target activated, first applied, and ended early for one target identity
INT-12	Pass	130 mg/dL target applied, persisted across cold relaunch, expired, notified, and restored permanent target
INT-13	Pass	Suspend/resume produced paused Home state, blocked meal, reminder, escalation, Pod-confirmed resume, and successful live step

Additional high-risk working observations:

- a canceled 4.25 U meal delivered 0.35 U, survived force-quit/relaunch, and was consumed exactly once at the next applicable step without full-dose reconstruction or a stuck meal state
- a long same-Pod outage reconciled 0.15 U of confirmed backup-basal delivery over bounded applicable steps and resumed live execution
- during suspend, no pump command was sent; resume recovery recorded a verified zero delivery delta for the bounded fallback-active interval before the next live command

Limitations And Promotion Requirements

- The working operator sheet lacks phone model/iOS and operator signature/date; this does not block its use as supporting formative evidence because no separate formal hardware-validation claim is made.
- Pod lot/expiry and CGM identity were not captured in this record.
- INT-07 and a same-run G7 replacement were not exercised.
- The recovery ZIP predates the later Temporary Target and suspend/resume exercises.
- A checker pass or operator report remains supporting evidence unless it is promoted through the study's controlled evidence process.
- Build 843 was designated for IDE submission on 2026-09-03. This supporting evidence was accepted under D08; no separate formal hardware-validation claim is made.

Conclusion

The evidence supports use of the recovery ZIP as the primary device-local technical record for its covered session interval, with Scout telemetry as corroborating backup. The evidence closes the engineering observation that Build 843 was installed and operated with a real Pod. Sponsor evidence acceptance is recorded under D08; Build 843 was approved for IDE submission on 2026-09-03. The no-separate-formal- hardware-validation claim boundary was clinically approved on 2026-08-27.

Build 843 Freeze Delta and Study-Build Designation

Updated: 2026-09-08

Field	Frozen source	Build 843 IDE submission build
Source identity	ide-software-freeze-2026-08-21 / 91c0e98a9bc9429a0486bebdebffc7d8dbbe300e	walkabout-candidate-2026-08-21-build-843 / 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Product version/build	Freeze source baseline	BionicLoop 1.0 (843)
Environment	Frozen engineering source	CC-DEV compiled Scout service environment; supportive service only, with no effect on local dosing
Archive SHA-256	Not applicable	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
Status	Exact-freeze execution complete	Build 843 approved for IDE submission 2026-09-03; evidence and traceability accepted 2026-09-07 EDT

1. Executable Delta Assessment

The source comparison from the frozen baseline to Build 843 identified these build-relevant changes:

Change	Assessment
Controlled app-workspace Package.resolved added at SHA-256 019b5768cfb4fc5c4c3c27106b7df9d60e0d7988eed34a3e75f4e523c625eded	Pins CryptoSwift 1.10.0, SlideButton, and SwiftCharts exactly for reproducible archive resolution
Config/Build/Base.xconfig and Debug.xconfig placed under source control	Configuration-identification correction; Build 843 values are retained at SHA-256 ed7d66f263f626e94e703538945ad63852b1792fbb739ee936c5e59ac38f9e2c and

Change	Assessment
	df4c78aa7ea8c5e7f0d2a4981e17dcbf1bc856ddce5671f918767f384a12a792
OmniBLE CryptoSwift requirement changed from a version range to exactly 1.10.0	Dependency-control change; no pump protocol or command behavior changed
OmniBLE test source changed from .bytes to .byteArray	Test compatibility only; no shipped implementation change

No BionicLoop app, BionicLoopCore, G7SensorKit, LoopKit, OmniBLE production Swift implementation, or Algo2015 C++ behavior changed between the frozen source and Build 843. The remaining comparison changes are documentation, evidence, scripts, and package records.

2. Verification Reuse Assessment

Evidence	Reuse assessment
Frozen algorithm, app, simulation, and TV-SEC-001 execution	Behaviorally applicable because no shipped runtime implementation changed; accepted for Build 843 under D08
Build 843 dependency checks	Directly applicable to Build 843
Build 843 OmniBLE cryptographic/session tests	Directly applicable to Build 843
Build 843 Core, canonical app, and security suites	Directly applicable to Build 843
Build 843 installed identity, real-Pod use, and exact-session recovery ZIP	Working/supporting evidence; accepted claim boundary must be recorded
RA-023 multi-device	Not established by current Build 843 working evidence. The September 3 adversarial review records CGM-screen manager-reset, concurrent-candidate, automatic-trigger-

Evidence	Reuse assessment
intended-sensor replacement	isolation, and first-reading-use limits. Sponsor/Clinical must decide whether unchanged Build 843 may proceed after objective testing and residual-risk acceptance or whether a corrected post-submission build is required before participant deployment; no discrimination claim is made before completion.

If a different archive, source commit, dependency resolution, signing identity, or environment is designated, this assessment must be repeated and the affected verification identified before participant installation.

3. Study-Build Designation

Build 843 was approved for IDE submission by controlled correspondence on 2026-09-03. Controlled correspondence references below replace handwritten signatures for this software-package record. The accepted D06 disposition retains the controlled RA-023 exercise as a pre-deployment condition.

Decision field	Sponsor entry
Designated participant study build (version/build)	BionicLoop 1.0 (843)
Source tag/commit	walkabout-candidate-2026-08-21-build-843 / 1455cda7ad5d3e164b4a955ea0e3bc725a9996e9
Archive SHA-256	80449b03c92dbc6aa21b5f255722522757f767ee84d303ab3af2b3b86f836ac0
Intended service environment	CC-DEV, internal study/tester environment
Signing/distribution identity	Retained in Appendix A10A release record
Accepted reused evidence	Completed at D08 closure
Required targeted reruns before use	Completed at D08 closure
Effective date	2026-09-03

Approval	Name/signature	Date
Sponsor/clinical approval	Camille Powe, MD; controlled correspondence	2026-09-03
Software technical review and final package approval	Edward R. Damiano, PhD; controlled correspondence	2026-09-07 EDT
Clinical/study operations / service environment	Tara Bresnahan, RN; controlled correspondence	2026-09-04

CC-DEV is compiled into Build 843 through `CC-DEV.xcconfig`. Scout is a supportive service and does not affect local dosing. A different service environment requires a new build identity and corresponding delta assessment. Participant-environment confirmation is closed under D12 in the IDE Submission Decision Register.

IDE Freeze Execution Report - 2026-08-21

Field	Value
Status	Engineering execution complete; Build 843 designated; evidence, residual-risk, cybersecurity, and final software-package approvals recorded
Owner	BionicLoop engineering
Baseline tag	ide-software-freeze-2026-08-21
Baseline commit	91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Last product-source commit in frozen baseline	ae00e754
Execution date	2026-08-21
Updated	2026-09-08

1. Scope

The four primary IDE software verification lanes were executed against the frozen baseline. Remaining sponsor decisions and physical observations are identified below.

2. Controlled Inputs

Execution used an isolated detached checkout of the annotated freeze tag.

Input	Identity
BionicLoop	91c0e98a9bc9429a0486bebdebffc7d8dbbe300e
Algo2015 nested source repository	b855f871c80e95d6f9226f54b0e60168ff603e36
Algorithm_2015_10_13.cpp	SHA-256 f2c61ddc46c45f40bd53712610e00060105d6fe4d912d25f61b34118dc10081e
Algorithm Interface .h	SHA-256 392970bd82f8cc5d978131e1118bb0288e95c57875e469092c8f6c9bcc80368f

Input	Identity
ringbuffer.h	SHA-256 869bacce22e01cb979bfeae4bf1f31c239b2f93689cc639d3cf305c46b6b8ab3
Base.xcconfig	SHA-256 ed7d66f263f626e94e703538945ad63852b1792fbb739ee936c5e59ac38f9e2c
Debug.xcconfig	SHA-256 df4c78aa7ea8c5e7f0d2a4981e17dcbf1bc856ddce5671f918767f384a12a792
Release.xcconfig	SHA-256 8c83ecc3acd2c23a2fcde93b2aa4e83329e9811d96a31b58bb47149cc55fb942
CC-DEV.xcconfig	SHA-256 9c50fa680863be12da1bd604695c6bfe0a2b6b79c5be44d519b077f777d34c80
Formal-run SwiftPM resolution	CryptoSwift 1.10.0 / f2a627b84c1ff96f21ac2fcb623ab36142dd5512; SlideButton 5eacebba4d7deeb693592bc9a62ab2d2181e133b; SwiftCharts c354c1945bb35a1f01b665b22474f6db28cba4a2; captured manifest SHA-256 019b5768cfb4fc5c4c3c27106b7df9d60e0d7988eed34a3e75f4e523c625eded
Toolchain	Xcode 26.6 (17F113), macOS 26.6.2 (25G83)

Algo2015, the Base/Debug configuration files, and app-workspace Package.resolved are local or ignored inputs in this repository. Their exact formal-run provenance and hashes are recorded rather than silently assumed. CYBER-DEV-001 covers the separate finding that archive 826 used a different CryptoSwift resolution.

3. Formal Lane Results

Lane	Result	Primary Evidence
STP-ALG-001	Behavioral pass; documentary deviation closed by companion	STR-ALG-001 run
STP-AUTO-001	App target pass; three UI-harness deviations closed by companion	STR-AUTO-001 run
STP-SIM-001	Pass	STR-SIM-001 run
TV-SEC-001	Pass for all 68 local controls defined by TV-SEC-001	STR-SEC-001 run
CYBER-DEV-001 impact companion	Pass with archive-matching CryptoSwift 1.9	dependency-impact run

Algorithm

- All behavioral suites passed with zero failed assertions.
- Coverage: Algo2015 96.12% line / 87.77% branch; bridge 90.27% line / 65.85% branch.

- Total-insulin host regression: 1/1 pass.
- Primary/safety framework identity report: pass.
- The original SA-006 run could not exact-match the freeze SHA because the immutable review-log entry uses `this commit`. The freeze linkage companion passed 6/6 with the literal SHA and requirement linkage, closing ALG-DEV-SA-006.

Automated App And UI

- App unit/integration target: 997 total, 996 passed, 1 intentional IFU reference-table generation helper skip gated by an export directory, 0 failed.
- The skip was `BionicLoopIFUReferenceExportTests.testExportWritesReferenceTablesWhenExportDirectoryIsSet`. It is an IFU document-generation helper gated by `IFU_EXPORT_DIR`, which is supplied only during controlled IFU reference-table generation; it does not omit a dosing or runtime requirement.
- UI target: 44 total, 41 passed, 3 failed.
- All three UI failures reproduced in isolation. Two are caused by an inconsistent profile seed. One compares accessibility frames for one-line and wrapped labels even though the production controls share one `HStack`.

Simulation

- Core, pod, alert, and real-engine simulation suites passed.
- `evaluation-summary.json` reports all TV-SIM-001..005 and TV-SIM-POD-001..004 true.

Cybersecurity Local Controls

- 68/68 focused tests passed.
- Participant Files-sharing/open-in-place plist keys are absent.
- Protected retention, export staging, backup exclusion, clinical share gate, and exact-session recovery archive behaviors passed all 68 tests defined by TV-SEC-001.
- Deferred cloud/auth/provider controls and commercial cybersecurity closure are not claimed by this result.
- The archive-matching companion passed 68/68 security tests defined by TV-SEC-001, 8/8 OmniBLE cryptographic/session vectors, 9/9 app unlock tests, and 19/19 core unlock vectors with no frozen product-source change.

4. Formal Deviations and Companion Status

ID	Description	Engineering Assessment	Disposition / Closure
ALG-DEV-SA-	Freeze review-log row is not	Documentary/tool-linkage defect;	Closed 2026-08-25 by freeze 6/6 linkage companion

ID	Description	Engineering Assessment	Disposition / Closure
006	literal-SHA machine-linkable	behavioral results unaffected	
AUTO-DEV-001	UI015/UI016 start from a Pregnancy seed while asserting Standard	Test-harness setup defect; production settings behavior not implicated	Closed 2026-08-25 by corrected 44/44 serial UI companion
AUTO-DEV-002	UI004a compares wrapped label accessibility frames	Test assertion does not measure the containing HStack	Closed 2026-08-25 by corrected 44/44 serial UI companion
CYBER-DEV-001	Package.resolved is ignored; archive 826 used CryptoSwift 1.9.0 while the freeze run resolved 1.10.0	Build-input identity mismatch, not a demonstrated product defect; the archive-matching impact companion passes	Closed 2026-09-03 by designation of Build 843 with controlled CryptoSwift 1.10.0 resolution; archive-826 history retained

The post-run composition review and evidence are in the freeze composition/manual-advisory companion. The matching-dependency verification is in the CryptoSwift 1.9 impact companion. The three closed non-product deviations are linked from the Formal Evidence Index. The original execution results remain unchanged.

5. Remaining Physical / Human Work

Status as of 2026-09-08

The numbered list below is retained as the original 2026-08-21 execution closeout record. Its actions now have these controlled dispositions:

Original item	Current disposition
1	Closed 2026-08-27 by Appendix A16: no separate formal hardware-validation or supplemental-alert claim is retained.

Original item	Current disposition
2	Superseded by the item 1 claim-boundary decision; the retained Build 843 real-Pod record remains supporting evidence in Appendices A10A-B.
3	Closed 2026-09-03 by Build 843 designation in Appendices A10C and A18; CYBER-DEV-001 is closed.
4	Closed for submission under D06 and D07; the controlled multi-G7 exercise and final deployment disposition remain required before participant deployment.
5	Closed by the approved IFU 1.17 and synchronized device-labeling record in Appendix A18 and D14.
6	Superseded by the supportive-Scout decision in Appendix A14 and closed cybersecurity disposition under D09; no retained live Scout round-trip is required for the submission claim.
7	Closed by the evidence-acceptance record under D08; the execution status itself remains unchanged.

The maintained closeout status, owners, and completion evidence are recorded in IDE_Freeze_Remaining_Work_Checklist_2026-08-21.md.

Engineering prepared a combined hardware/alert operator worksheet, retained as an internal supporting record rather than a separate formal claim.

The remaining actions are:

1. The sponsor chooses whether STP-HW-001 and supplemental STP-ALERT-001 remain claimed or are explicitly deferred.
2. If retained, an operator executes/signs the packet on the exact frozen build and attaches device, build, time, and saline-Pod evidence.
3. The sponsor representative dispositions CYBER-DEV-001; a passing archive-matching companion is available for review.
4. Sponsor/clinical owners resolve the remaining residual items identified in the freeze disposition memo and any retained hardware-only claims.
5. Complete controlled IFU review/approval and required signatures.
6. Attach the live retained-sequence phone-to-Scout observation and record acceptance of the controlled manual advisory method, or approved scanner artifacts if a scanner run is required.
7. The sponsor representative accepts the formal evidence bundles and dispositions the applicable RTM rows while preserving explicit deferrals.

6. Freeze Control

No product source was changed during formal execution. Any product/build-input change requires a new freeze identity and impact-based rerun. Documentation or test-harness companion changes must not be represented as results from the original immutable tag.

Post-Approval Software Change-Control and Regulatory-Classification Procedure

Field	Value
Status	Controlled procedure for sponsor and Regulatory approval
Date	2026-09-03
Applies after	IDE approval and controlled study deployment
Final classification owner	Sponsor/Regulatory lead
Regulatory basis	21 CFR 812.35 and applicable FDA guidance

1. Purpose and Authority

This procedure defines how software, configuration, labeling, cybersecurity, telemetry, device-integration, and investigational-plan changes are assessed, verified, approved, reported, and traced after IDE approval. It does not make a regulatory decision. The Sponsor/Regulatory lead owns the final classification, with FDA and IRB interaction as applicable. When classification is uncertain, the Sponsor/Regulatory lead should consult the FDA reviewing division.

Engineering effort, implementation time, or schedule impact does not determine regulatory significance. A UI change is not automatically minor.

2. Required Assessment for Every Proposed Change

1. change description, rationale, affected versions, and deployment intent
2. engineering delta identifying affected code, configuration, interfaces, data, UI, alerts, devices, algorithms, dependencies, and cybersecurity
3. affected RA-*, SRS-*, SDD-*, TV-*, IFU, protocol, consent, monitoring, data-management, and cybersecurity artifacts
4. new/changed hazards, controls, residual risk, subject impact, and study-data impact
5. objective regression and verification plan
6. labeling, training, protocol, consent, endpoint, monitoring, IRB, and cybersecurity assessment
7. Sponsor/Regulatory classification and implementation authorization
8. deployment and version-to-subject/data traceability plan

No change may be distributed to an investigator or subject before its required engineering, independent-review, sponsor, FDA, and IRB gates are complete, except for the narrow emergency exception in section 3.3.

3. Classification Categories

3.1 Internal or Nondeployed Change

Development experiments, test-harness changes, documentation drafts, and internal tools may remain outside the deployed IDE configuration only when no subject/investigator device, production service, labeling, protocol, relied-upon study-data system, or subject data is changed. Source control must identify the work as nondeployed. Before any later deployment, the complete change is reclassified against the approved IDE and current study state.

3.2 Approved Configuration Within an Approved Range

A clinician selecting a glucose target, meal-upfront percentage, TMAX, or Temporary Target already allowed by the approved software and protocol is a configuration action, not a new software version. Record the subject/device, old/new value, authorized operator, rationale, time, and study phase. A new range, default, decision rule, or permitted operator is a change requiring classification under sections 3.3 through 3.7.

3.3 Emergency Deviation to Protect a Subject

When a deviation from the investigational plan is necessary to protect the life or physical well-being of a subject in an emergency, it may be implemented without prior FDA approval under 21 CFR 812.35(a)(2). The sponsor must notify FDA within five working days after learning of the deviation, and the investigator must promptly notify the sponsor and reviewing IRB as required. Document affected subjects/data, risk, corrective action, and prevention. This exception is not a route for planned changes.

3.4 Developmental Device Change Eligible for Five-Day Notice

Under 21 CFR 812.35(a)(3)(i), a developmental device/software change may be implemented without prior FDA approval and reported within five working days after making the change only when the sponsor documents credible information that the change:

- was made in response to information gathered during the investigation
- does not constitute a significant change in design
- does not constitute a change in the device's basic principles of operation

The notice describes the change and summarizes the supporting information. Five-day notice is not prior approval. Required design-control, risk, verification, deployment, and IRB assessments still apply.

3.5 Qualifying Protocol Change Eligible for Five-Day Notice

Under 21 CFR 812.35(a)(3)(ii), a protocol change may be implemented without prior FDA approval and reported within five working days after making the change only when the sponsor documents credible information that the change does not affect:

- the validity of the data or information resulting from completion of the approved protocol
- the relationship of likely subject risk to benefit
- the scientific soundness of the investigational plan
- the rights, safety, or welfare of human subjects

Five-day notice is not prior approval. IRB notification or approval remains a separate required assessment.

3.6 Minor Change Reported in the Annual Progress Report

The annual-report route in 21 CFR 812.35(a)(4) is limited to minor changes to:

- the purpose of the study
- risk analysis
- monitoring procedures
- labeling
- informed-consent materials
- IRB information

The sponsor must document that the change does not affect scientific soundness, data validity, subject risk-benefit, or subject rights, safety, or welfare. This category does not create a general annual-report route for minor software or UI changes.

3.7 Change Requiring Prior IDE Supplement Approval

Prior FDA approval under 21 CFR 812.35(a)(1), and IRB approval when applicable, is required unless another provision above applies. Escalate changes that may affect scientific soundness, data validity, subject rights/safety/welfare, risk-benefit, significant design, basic principles of operation, or the approved investigational plan.

Potential examples include algorithm/controller changes; dosing limits, safeguards, reconciliation, or backup-basal policy; device command semantics; safety-critical alert behavior; clinical-unlock authority; cybersecurity trust boundaries; relied-upon telemetry completeness; target ranges; or meal rules. Examples are screening prompts, not automatic classifications.

4. Study-Relevant Assessment Examples

Change	Required assessment
UI wording or display	Meaning, action, urgency, interpretation, accessibility, training, labeling, protocol, data validity, and use-related risk
Alert trigger, severity, suppression, or recovery action	Delayed/missed response, notification behavior, IFU/protocol impact, and subject risk
Clinical unlock or access control	Unauthorized configuration, provisioning/recovery, cybersecurity, and study operations
G7 or DASH integration	Wrong-device association, stale state, command/delivery semantics, supplier controls, and bench/hardware regression
Algorithm or dosing behavior	Basic principle of operation, risk-benefit, protocol/endpoints, and full algorithm/runtime/device regression
Dosing safeguard or reconciliation	Over/under-delivery, exactly-once accounting, relaunch/replacement, simulation, and hardware evidence
Cybersecurity control or dependency	Threat model, SBOM/advisory evidence, trust boundary, authentication, local data, and clinical availability
Telemetry or backend	Whether Scout is supportive or relied upon, compatibility, sequence/data loss, source attribution, Part 11 scope, and endpoint analysis
Internal/test-only work	Proof that no deployed binary, configuration, evidence interpretation, relied-upon system, or subject data changed

5. Verification and Approval

Before authorized implementation or distribution, as applicable:

- update requirements, design, risk, traceability, cybersecurity, and labeling
- complete regression proportional to risk and all affected formal lanes
- use simulation for reproducible cases and hardware for claims that cannot be adequately simulated
- complete independent technical review and sponsor evidence acceptance
- document Sponsor/Regulatory classification, FDA reporting route, and IRB action
- create a new immutable software/configuration identity

6. Version-to-Subject and Version-to-Data Traceability

For every distributed build or approved configuration change, retain source tag and SHA; archive/build number; archive checksum; algorithm/dependency identities; signing/distribution identity; subject/device/install assignment; distribution, activation, and retirement times; configuration history; telemetry/data schema and server contract when relied upon; study-data interval; and rollback, quarantine, or migration action. Study data must remain attributable to the software and configuration that generated it.

7. IRB, Labeling, Protocol, and Reporting Assessment

Every classification record states whether the change affects protocol, informed consent, IFU/labeling, training, monitoring, data management, statistics, investigator instructions, or subject safety. The Sponsor/Regulatory lead determines the FDA route and resolves uncertainty with the reviewing division. The sponsor and IRB determine IRB notification or approval.

8. Decision Record Template

This template is completed for each applicable software change. Blank fields below are intentional template fields, not missing approvals for the current submission baseline.

Field	Required entry
Change identifier and versions	_____
Deployed to subjects or relied-upon study systems?	_____
Engineering delta/risk/regression assessment	_____
Protocol/labeling/consent/training/data impact	_____
Regulatory category and rationale	_____
FDA report/supplement and date	_____
IRB action and date	_____
Sponsor/Regulatory approval	_____
Independent technical review / evidence acceptance	_____
Deployment and subject/data trace record	_____

9. Regulatory References

- 21 CFR 812.35, Supplemental applications: <https://www.ecfr.gov/current/title-21/chapter-I/subchapter-H/part-812/subpart-B/section-812.35>

- FDA, IDE Application: <https://www.fda.gov/medical-devices/investigational-device-exemption-ide/ide-application>
- FDA, Changes or Modifications During the Conduct of a Clinical Investigation: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/changes-or-modifications-during-conduct-clinical-investigation-final-guidance-industry-and-cdrh>
- FDA, IDE Reports: <https://www.fda.gov/medical-devices/investigational-device-exemption-ide/ide-reports>

IDE Sponsor and Nonsoftware Ownership Map

Field	Value
Status	Gap and ownership map; does not assert completion
Regulatory basis	21 CFR 812.20 and 812.25; FDA IDE application guidance
Date	2026-09-04

Purpose

This map distinguishes the engineering-owned software package from sponsor, clinical, site, data-management, manufacturing, monitoring, and regulatory materials needed for a complete IDE. `Not identified in controlled software records` means that the software package does not contain controlled evidence of completion; it does not mean the artifact does not exist elsewhere.

Ownership Map

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
Sponsor identity and correspondence	Sponsor/Regulatory	IDE cover letter, sponsor contact, correspondence authority	Not identified in controlled software records	None identified	Controlled sponsor/correspondent record
Report of prior investigations	Sponsor with Clinical/Engineering	Comprehensive clinical, animal, laboratory, adverse, and publication report	Software portion available; complete report not evidenced	Software Prior Testing Summary; formal software evidence	Nonsoftware and clinical prior-investigation package; GLP statements where applicable
Investigational plan	Sponsor/Clinical	Approved investigational plan	Final Protocol Version 1.0 dated 1 September 2026	Software consistency crosswalk;	Nonsoftware submission administration is

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
purpose and duration			reviewed for software consistency	Software Submission Core	outside this software map
Clinical protocol and analysis	Clinical PI/Protocol lead, Biostatistics	Approved protocol, statistical analysis, case-report forms	Final protocol is materially consistent with Build 843	Software consistency crosswalk	Statistical and CRF package are outside the software package
Comprehensive risk analysis and risk minimization	Sponsor/Clinical with Engineering and sponsor-designated review approver	Clinical and device risk analysis	Software risk analysis available; complete IDE risk package not evidenced	Risk Analysis, human-factors summary, anomaly register	Sponsor residual-risk acceptance and nonsoftware/clinical risk integration
Device description and anticipated changes	Engineering, Device/Manufacturing, Sponsor	Complete device description and principles of operation	Software description available; complete device package not evidenced	Software Submission Core; SDD	Hardware, accessories, supplier device descriptions, anticipated change plan
Monitoring procedures and monitor identity	Sponsor/Clinical Operations	Monitoring plan, monitor names/addresses, escalation and source-data review	Not identified in controlled software records	Protocol references monitoring generally	Controlled monitoring plan and assigned monitors
Additional study records and reports	Sponsor/Data Management/Clinical	Record/report inventory beyond 21	Partially described; controlled inventory not identified	Protocol data sections; telemetry scope decision	Data-management plan, source-of-truth decisions, report schedule

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
	Operations	CFR 812 Subpart G			
Manufacturing, processing, packing, storage, installation controls	Manufacturer/Sponsor with designated review approver	Manufacturing and device-control description	Not identified in controlled software records	Software configuration identification only	Device manufacturing, packaging, storage, installation and quality controls
Device accountability and distribution	Sponsor/Clinical Operations/Sites	Shipment, custody, assignment, return, destruction, version-to-subject records	Software version-to-subject requirement defined; execution records not identified	Change-control procedure; app telemetry concepts	Controlled accountability logs and site procedures
Investigator agreements, qualifications, and list	Sponsor/Clinical Operations	Signed agreements, CVs, financial disclosure, investigator list	Not identified in controlled software records	None identified	Complete signed investigator package
IRB information and approval	Sponsor/Sites/IRB	IRB list, chairpersons, submissions, approvals, continuing review	Not identified in controlled software records	None identified	Controlled IRB approvals and correspondence
Other participating institutions	Sponsor/Clinical Operations	Institution list and responsibilities	Not identified in controlled software records	Protocol identifies MGB context	Controlled complete site/institution list

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
Device charges/commercialization statement	Sponsor/Finance/Regulatory	Charge statement and noncommercialization rationale	Not identified in controlled software records	None identified	Sponsor decision and submission content
Labeling	Sponsor/Regulatory/Clinical with Engineering and designated review approver	21 CFR 812.5 labeling, IFU, device labels, warnings	IFU-BL-001 revision 1.17 and synchronized final device labeling are approved and verified	IFU-BL-001 revision 1.17; September 2026 approval record; synchronized 008_Device_Labeling.pdf; app investigational banner; software consistency crosswalk	Assemble the controlled label and IFU attachments in the final submission
Informed consent and participant materials	Sponsor/Clinical/IRB	Approved consent and all participant-facing materials	Final consent software disclosure is consistent with Build 843	Software consistency crosswalk	General consent and IRB administration are outside the software package
Training	Clinical Operations/Sites	Trainer qualifications, curriculum, competency and completion records	Procedure described; controlled records not identified	Protocol training text; IFU	Approved curriculum, competency criteria, completed records

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
Clinical data management	Data Management/Biostatistics/Sponsor	Data-management plan, data sources, reconciliation, retention, analysis transfer	Official sources identified: Dexcom Clarity for CGM outcomes, BionicLoop exported CSV for insulin-delivery outcomes; Scout supportive/optional	2026-08-27 and 2026-08-31 clinical approval record; protocol data sections; local protected evidence and recovery export; Scout support controls	Database controls, reconciliation, retention, analysis-transfer plan, and controlled incorporation into protocol/data-management documents
Safety and progress reporting	Sponsor/Medical Monitor/Clinical Operations	UADE, deviation, progress, final, IRB, and FDA report procedures	EDC/eCRF identified as the official safety-event record; proposed wording allows participant report/assessment and medical-record review as source information, with investigator causality/relatedness determination; operational procedure remains pending	2026-08-27 and 2026-08-31 clinical correspondence; software alert/evidence records	Controlled reporting procedures, owners, timelines, EDC/eCRF controls, causality/relatedness process, and completed reports
Clinical monitoring and medical oversight	Medical Monitor/Clinical lead	Safety monitoring, escalation, review and stopping procedures	Partly described in protocol; controlled operational package not identified	Protocol source	Approved monitoring and medical-oversight records
Environmental categorical exclusion and other FDA-requested	Sponsor/Regulatory	Applicable submission statement and requested information	Not identified in controlled software records	None identified	Regulatory determination and submission content

IDE element	Primary owner	Expected source artifact	Current status	Controlled software evidence	Missing evidence or decision
information					

Submission Control

The sponsor/Regulatory lead owns the final IDE completeness assessment. The software package closes only the engineering-owned portion and must not be used to infer completion of nonsoftware elements.

Primary Regulatory Sources

- FDA, IDE Application: <https://www.fda.gov/medical-devices/investigational-device-exemption-ide/ide-application>
- FDA, IDE Approval Process: <https://www.fda.gov/medical-devices/investigational-device-exemption-ide/ide-approval-process>

IDE Clinical Policy Confirmation - 2026-08-21

Field	Value
Status	Clinical confirmation complete
Requester	BionicLoop Software Developer
Clinical reviewer	Camille Powe, MD
Request date/time	2026-08-21 11:11 EDT
Confirmation date/time	2026-08-21 11:38 EDT
Recorded in package	2026-08-25
Last updated	2026-09-03

Confirmed Decisions

The clinical reviewer replied "Confirmed" to the request containing all three statements below:

1. Backup-basal reconciliation distributes the accepted delivery total across the applicable five-minute steps according to the backup-basal schedule. Individual steps are rounded to 0.01 U while preserving the accepted total.
2. Backup basal may activate when current CGM freshness cannot be established, as well as when CGM data is confirmed to be stale.
3. The participant weight-entry range is 50...500 lb inclusive (22.7...226.8 kg). Entries outside this range prevent the algorithm from starting. The algorithm also applies a final 20...230 kg safety limit.

Source Evidence

- Retained source: clinical confirmation email screenshot
- SHA-256: 026a4da0569c56ff74992ea41bfc99a7bb8ab92f48c88f7402875ff5147b8a2a
- The retained image is a byte-for-byte copy of the supplied email screenshot.

Disposition Boundary

This record closes the outstanding clinical-confirmation request for schedule-weighted fallback quantization, RES-004, and the weight-entry range. It does not change product behavior or formal execution results.

Final IDE submission-build approval was recorded on 2026-09-03. Organizational residual-risk approval and submission signatures are sponsor, Clinical, and Regulatory activities. They are not additional BionicLoop engineering implementation deliverables.

IDE Cloud Telemetry Submission-Scope Decision

Field	Value
Status	Clinically approved: Scout supportive
Decision owners	Sponsor representative with clinical-investigator input
Date opened	2026-08-21
Package position recorded	2026-08-25
Clinical approval and source clarification	2026-08-27
Updated	2026-08-27

Issue

Prior software scope language deferred cloud and device-to-cloud verification, while the Cybersecurity Plan described secure cloud telemetry as a primary study transport. Those statements could not govern the same package.

The software package uses the supportive Scout scope. Clinical approval and the authoritative clinical data sources were recorded on 2026-08-27.

For the IDE software package, Scout is classified as supportive. The device-local clinical-gated recovery ZIP is the primary session-reconstruction record used for software evidence, and Scout telemetry is corroborating backup for monitoring, correlation, and gap investigation. Temporary Scout delay or loss does not affect local algorithm execution or dosing.

The clinical correspondence identifies the official study sources in the decision table below.

Approved Scope

If Scout is not relied upon for safety decisions or required endpoint data:

- describe Scout as supportive monitoring and engineering/operational evidence
- identify the authoritative sources for dosing, safety, and endpoint data in the protocol, data-management plan, and submission
- state that temporary Scout delay or loss does not affect local dosing
- retain local protected step evidence and recovery export as support controls

- defer broader cloud, provider/authentication, Part 11, and commercial closure with explicit rationale
- ensure the IFU and monitoring plan do not instruct staff to depend on Scout for a safety action that lacks another authoritative path

Approved sources and purposes:

Required identification	Sponsor response
Software verification/session reconstruction	Protected local step archive and clinical-gated exact-session recovery ZIP; reconcile to pump records where the protocol requires
Official CGM outcome source	Dexcom Clarity
Official insulin-delivery outcome source	BionicLoop app exported CSV file
Official adverse-event and other safety-information source	Electronic case-report forms entered by study staff
Scout support purpose	Study monitoring, troubleshooting, corroborating software evidence, sequence/gap review, and recovery assistance; not a local dosing dependency

Decision Record

Field	Decision
Decision / date	Scout supportive; engineering package position recorded 2026-08-25 and clinically approved 2026-08-27
Rationale	The controller retains protected local step evidence and a clinical-gated exact-session recovery export; local dosing does not depend on Scout availability. Scout remains valuable corroborating evidence and monitoring support.
Authoritative data sources	Software evidence: local protected records/recovery ZIP and formal verification bundles. Official study outcomes: Dexcom Clarity for CGM and the BionicLoop exported CSV for insulin delivery. Adverse events and other safety information: staff-entered electronic case-report forms.
Required Scout contract/evidence	No Scout round trip is required as the primary software-verification record. Retain available sequence-correlated Scout data as corroborating backup and investigate material gaps.
Deferred cloud scope	Commercial cloud, broader provider/authentication, Part 11, and production-role closure remain deferred unless the relied-upon study scope is later widened.

Field	Decision
Approval reference / date	Camille Powe, MD email, Re: A few final software decisions for the IDE package, 2026-08-27
Controlled decision record	Clinical Approval of Software IDE Decisions

Sequence-correlated phone-to-Scout observations may be retained as operational transport evidence and used to investigate gaps. They are not required to establish local dosing execution or reconstruct a software session when the protected local archive and exact-session recovery ZIP are available.

IDE Protocol Fingerstick Consistency Issue

Updated: 2026-09-04

Field	Value
Status	Closed in the final protocol and consent document set
Source	Clinical Protocol BLS Clean-final.docx; <i>Optimizing Automated Insulin Delivery to Meet Pregnancy Glycemic Targets</i> , Version 1.0, version date 1 September 2026, 41 pages; SHA-256 4bfb09542230c94c00dc5b5331a969657b5961b1c4bbd5a2759519a00f71d264
Affected software areas	Manual BG, CGM outage, algorithm mode, alerts, transition behavior
Clinical approval	Camille Powe, MD approved the distinct safeguarded fingerstick direction on 2026-08-27 and approved implementation of the related protocol/consent recommendations on 2026-08-31

Historical Protocol Discrepancy

The June protocol stated that entered fingerstick BG measurements would be treated as Dexcom CGM data. That statement did not describe the implemented software accurately.

Implemented Behavior

The frozen software does not treat a fingerstick value as an ordinary Dexcom CGM sample. It uses a distinct BG-based closed-loop path with controls that include:

- manual range and completeness validation
- explicit exact-value user confirmation
- step-scoped pending validity and exactly-once consumption
- BG-specific timing and due/overdue scheduling during sustained CGM outage
- BG-specific alerting and Home presentation
- shared input to both algorithm copies only on the qualifying step
- explicit transitions between CGM-based, BG-based, and forced-open behavior
- no step-0 BG rescue in the frozen baseline

The historical wording therefore overstated equivalence between a fingerstick BG and Dexcom CGM data.

Verified Final Protocol Direction

If Dexcom CGM data are unavailable, a participant may enter and confirm a fingerstick BG. BionicLoop processes the value through a distinct BG-based closed-loop mode with BG-specific timing, validity, alert, and transition safeguards. The value is available to the next eligible algorithm step, is consumed once, and may result in automated insulin delivery. The Dexcom G7 should be restored or replaced as soon as possible.

The final protocol describes the distinct safeguarded fingerstick mode, next eligible step, and exactly-once consumption consistently with this direction. The clinical protocol remains controlled outside this software repository.

Required Disposition

Decision	Owner/status
Confirm the intended clinical role of fingerstick BG during CGM outage	Approved by Camille Powe, MD on 2026-08-27
Approve direction to replace the Dexcom-equivalence statement with wording consistent with implemented behavior	Approved by Camille Powe, MD on 2026-08-27
Confirm the final consent and IFU provide participant-facing CGM-outage instructions	Closed by the final consent document and current IFU revision 1.17

The software consistency issue is resolved in the final protocol, consent, and IFU document set. The approved clinical direction is retained in the Clinical Approval of Software IDE Decisions. The broader June-document review and proposed wording are recorded in the Software Consistency Crosswalk.

Clinical Approval of Software IDE Decisions - 2026-08-27 and 2026-08-31

Field	Value
Evidence type	Controlled correspondence decision record
Respondent	Camille Powe, MD
Initial response date/time	2026-08-27 10:30 AM EDT (14:30 UTC)
Follow-up response date/time	2026-08-31 8:00 AM EDT (12:00 UTC)
Email subject	Re: A few final software decisions for the IDE package
Source retention	Original email retained in project correspondence outside the software repository
Product impact	Documentation and submission-scope disposition only; no software behavior change

Approval Statement

The 2026-08-27 response approved the seven emailed software/IDE statements and clarified the official study-data and safety-report sources. The 2026-08-31 follow-up instructed the study team to make all eight software-consistency changes recommended in the 2026-08-28 review and return tracked protocol and consent updates for review. The follow-up therefore records clinical agreement with the change direction; it does not establish that the controlled protocol, consent, or labeling has already been revised or approved for submission.

Approved Decisions

- Algorithm start input:** The algorithm requires a Dexcom reading when it is first started. A fingerstick cannot replace that first reading.
- CGM gaps:** During a connection outage, the app does not synthesize missing historical Dexcom readings. It records them as unavailable and accounts for insulin delivered during the outage.
- Clinical Settings access:** Study staff enter Clinical Settings during initial setup. Later edits require a single-use clinical unlock code. This is an investigational study control, not a commercial hospital user-account system.

4. **BionicScout scope:** Scout supports monitoring and troubleshooting, but the phone continues dosing without it. Scout is not the official source of study outcomes or safety reporting.
5. **Physical-use claim:** The package will describe extensive automated testing and formative real-Pod use supported by retained records. It will not claim a separate formal hardware-validation study.
6. **Human-factors claim:** The package will describe engineering UI testing, formative real-device testing, and clinical review. It will state that no separate formal participant usability study was conducted.
7. **Fingerstick protocol wording:** The protocol should describe the app's distinct safeguarded fingerstick process rather than state that an entered fingerstick is treated as Dexcom CGM data.

The approved claim boundary means STP-HW-001 is not retained as a separate formal hardware-validation claim. The approved description of app usability, formative real-device testing, and clinical review likewise does not retain supplemental STP-ALERT-001 as a separate operator-observed formal study claim. Existing automated alert verification and formative real-Pod records remain supporting evidence.

Approved Clinical-Document Change Directions

The 2026-08-31 follow-up approves implementation of these eight recommendations from the software consistency review:

1. describe fingerstick BG as a distinct safeguarded input mode, including the related meal-availability and participant-disclosure language during CGM interruption;
2. describe the Standard and Pregnancy permanent targets, Pregnancy Temporary Targets, automatic return, unchanged Pregnancy safety/backup basis, and the Standard-to-Pregnancy 90% meal-setting proposal with clinical review before save;
3. align the protocol with the 50 through 500 lb inclusive app-entry range and the algorithm's 20 through 230 kg final safety range;
4. replace Phase 3 "factory reset" wording with the Same-Participant Reset workflow and reserve New Participant Reset for phone reassignment;
5. describe G7 replacement as finishing startup and providing readings rather than requiring routine calibration;
6. use the approved official-source hierarchy below and retain the recovery ZIP and Scout as supporting sources;
7. describe the evidence as risk-based software verification and engineering representative-use testing under simulated and real-device conditions, without claiming a separate formal hardware-validation or summative participant-usability study; and

8. complete the software-labeling set with the designated build, approved IFU revision, study-phone label where applicable, and in-app investigational-use banner specimen.

Clarified Official Study Sources

Study information	Official source
CGM outcome data	Dexcom Clarity
Insulin-delivery outcome data	BionicLoop app exported CSV file
Adverse events and other safety information	Electronic case-report forms entered by study staff

Tara Bresnahan, RN separately proposed that the controlled study documents state that adverse events and other required safety events are entered directly into the EDC by study staff; source information may come from participant report or assessment and medical-record review; and the investigator determines causality and relatedness to the study device and protocol. This is retained as document-update input pending incorporation and approval in the controlled protocol, data plan, or safety-reporting procedure.

The protected device-local recovery ZIP remains the primary software session-reconstruction record for the covered interval. BionicScout remains supportive monitoring and corroborating evidence. These software-evidence roles do not replace the official clinical study sources above.

Remaining Administrative Action

The correspondence did not identify the person authorized to give final sponsor or submission acceptance. The study team must still designate that role and retain final controlled approvals. Revised protocol, consent, and labeling files must implement the approved directions and receive their applicable document-control and FDA/IRB dispositions.

Insulin Delivery CSV Data-Management and Chain-of-Custody Procedure

Field	Value
Official source covered	BionicLoop app exported insulin-delivery CSV
Supporting source	Clinical-gated exact-session recovery ZIP containing the CSV and matching algorithm files
Scout role	Corroborating monitoring and gap investigation; not the official insulin-delivery outcome source
Status	ZIP/CSV collection, transfer, and storage approved 2026-09-03; detailed operational procedure retained for controlled use
Date	2026-09-08

1. Purpose and Scope

This procedure controls collection, integrity review, transfer, retention, and use of the BionicLoop insulin-delivery CSV identified as the official insulin-delivery outcome source. It applies to every participant phone and algorithm session used for study data.

Adverse events and other required safety events are outside the CSV outcome record. The proposed study-document wording identifies the EDC/eCRF entered by study staff as the official safety-event record. Source information may come from participant report or assessment and medical-record review, and the investigator determines causality and relatedness to the study device and protocol. The study's controlled safety-reporting procedure remains the authority for those operations.

2. Roles

Function	Responsibility
Study site/clinical operator	Export the correct session, record the source identity and time, and transfer the original package without editing it
Data Management	Receive, hash, inventory, validate, query discrepancies, preserve originals, and produce analysis copies
Software Developer	Maintain the export definition, assist with technical reconstruction, and assess software anomalies without modifying source records

Function	Responsibility
Sponsor	Approve cadence, destination, retention, access, reconciliation rules, and final data disposition

3. Collection Preconditions

Before export, the operator records:

- participant/subject identifier
- study phone/device identifier
- installed BionicLoop version/build and study-build designation
- algorithm session start time or epoch identity
- export date/time and time zone
- operator identity
- reason for collection: scheduled, treatment-period end, device transition, anomaly investigation, or study closeout

The exact routine export cadence and required treatment-period checkpoints must be defined in the approved protocol or data-management plan before first participant use.

4. Export and Original Record

1. Use the clinical-gated Recent Dose Steps export.
2. Select the intended algorithm session by its displayed start date/time.
3. Export the generated ZIP. The ZIP must contain the session CSV and may contain only the matching legacy algorithm artifacts; partial algorithm-file presence does not invalidate a complete CSV.
4. Do not open and resave, filter, rename internal files, or edit the original CSV.
5. Calculate SHA-256 for the original ZIP immediately after receipt in the controlled study repository.
6. Record the hash, original filename, byte size, source details, operator, collection time, receipt time, and controlled storage locator in the data inventory.

An empty archive, missing CSV, wrong participant/session, or archive that contains files from more than one algorithm session is not accepted as an official source record and requires a data query.

5. Integrity and Completeness Review

Data Management verifies and records:

- ZIP integrity and reproducible SHA-256

- filename/session identity against the operator record
- CSV header/schema identity and readable row structure
- expected session start/end coverage
- monotonic algorithm-step progression, with replay/attribution rows interpreted using their evidence fields rather than wall-clock order alone
- duplicate rows, missing ranges, impossible values, or unexplained session changes
- relevant reconciliation/evidence labels for canceled, interrupted, assumed, or fallback delivery
- comparison to pump records when required by the protocol/data plan

Observed gaps or discrepancies create a controlled data query. The original file remains immutable. Any corrected or derived dataset receives a new identifier, transformation record, preparer/reviewer, date, and link to the original hash.

6. Recovery and Corroboration

If the official CSV cannot be exported or appears incomplete:

1. preserve the phone and original session state where possible;
2. repeat the exact-session recovery export without resetting participant data;
3. use matching algorithm files, retained local evidence, pump history, and sequence-correlated Scout records to investigate the gap;
4. document every source and assumption in the data query;
5. do not silently substitute Scout telemetry for the official CSV.

7. Transfer, Access, and Retention

- Transfer only through the sponsor-approved protected study-data channel.
- Limit access to authorized study, Data Management, technical, and regulatory personnel with a documented need.
- Retain the immutable original ZIP, its hash/inventory record, data queries, and derived-dataset lineage for the period defined by the protocol, sponsor policy, and applicable IDE record-retention requirements.
- Record any copy, movement, or disposition that changes the controlled storage locator.

8. Approval Record

Approval	Controlled approval reference	Date
ZIP/CSV collection, transfer, and storage approach	Camille Powe, MD, controlled correspondence in the Bionic Loop Software – Final Approvals thread	2026-09-03
Software-package technical review	Edward R. Damiano, PhD, final software-package approval in the Bionic Loop Software – Final Approvals thread	2026-09-07 EDT

These controlled references replace handwritten signatures for this software-package procedure. Study operations retain responsibility for following the protocol/data-plan cadence and sponsor-controlled storage rules.

Final Software Approvals - September 2026

Field	Value
Evidence type	Controlled correspondence decision record
Respondents	Camille Powe, MD; Tara Bresnahan, RN; Edward R. Damiano, PhD
Response dates	2026-09-03 through 2026-09-07 EDT
Email subject	Bionic Loop Software - Final Approvals
Source retention	Original email retained in project correspondence outside the software repository
Product impact	Submission designation and document/process approval only; no software behavior change

Approved Items

The September 3 correspondence records approval of:

1. BionicLoop 1.0 (843) for IDE submission;
2. IFU-BL-001, revision 1.16, for IDE submission; and
3. use of the recovery ZIP for CSV collection, transfer, and storage.

The approval closes the software-build and IFU selection decisions and the study-team approval of the ZIP/CSV operating approach. The approved study-data hierarchy remains Dexcom Clarity for CGM outcomes, the BionicLoop exported CSV for insulin-delivery outcomes, and staff-entered electronic case-report forms for adverse events and other safety information.

IFU Revision 1.17 Direction - 2026-09-04

Follow-on correspondence in the same email thread records:

- Tara Bresnahan, RN supported applying the proposed IFU corrections; and
- Camille Powe, MD directed the team to make the changes and submit IFU-BL-001 revision 1.17.

The approved revision corrects two Clinical Unlock section references, clarifies Pregnancy configuration terminology, records the revision 1.16 approval, replaces an internal build-tool instruction with controlled-document language, uses plain punctuation, and documents the existing built-in NOT FOR HUMAN USE badge displayed when no remote banner configuration is available. These are labeling and document-control corrections only; they do not change Build 843 software behavior.

Final Decision-Register Approvals - 2026-09-04 and 2026-09-07

In the same controlled email thread:

- Tara Bresnahan, RN accepted D06, including the documented provisional residual-risk rating and completion of the controlled multi-G7 exercise before participant deployment rather than before IDE submission;
- Tara Bresnahan, RN accepted D12, confirming CC-DEV as Build 843's supportive Scout service environment;
- Tara Bresnahan, RN authorized Edward R. Damiano, PhD to provide final software-package approval under D11; and
- Edward R. Damiano, PhD accepted the documented residual risks and known limitations (D07), Build 843 verification evidence and traceability (D08), and the documented cybersecurity scope and controls (D09). He also accepted the D11 designation and signed off on the final software package.

The source messages are dated September 4 and September 7, 2026 EDT. Original email is retained in project correspondence outside the software repository.

Final Device Labeling - 2026-09-04

The study-team submission folder contains 008_Device_Labeling.pdf, a 3-page, 230,910-byte final labeling file identifying BionicLoop 1.0 Build 843 and IFU-BL-001 revision 1.17 dated 2026-09-04. Software review verified SHA-256

561c4cd26171888454c27d2350162680a867190754a448fd4096d1dd8b10fe0c. This closes the document-handoff requirement tracked as D14.

Remaining Operational Work

The accepted D06 disposition retains the controlled multi-G7 exercise as a pre-deployment condition. Final IDE assembly, transfer, and submission remain under study-team control. Neither item changes the approved software package or reopens a software-package decision.